AD-A038 400   MINNESOTA UNIV MINNEAPOLIS MANAGEMENT INFORMATION SY--ETC F/G 9/2
AN AUTOMATED ANNOTATED BIBLIOGRAPHY ON THE SPECIFICATION OF INF--ETC(U)
OCT 76 G C EVEREST, O BRAY, I VALTERS   N00167-76-M-8441

UNCLASSIFIED   MISRC-TR-77-01   NL

1 OF 2
AD
A038400

12

Report Number: MISRC-TR-77-01

AN AUTOMATED ANNOTATED BIBLIOGRAPHY
ON THE SPECIFICATION OF INFORMATION SYSTEM REQUIREMENTS

Management Information Systems Research Center
Graduate School of Business Administration
University of Minnesota
Minneapolis, MN 55455

1976 October 11

FINAL REPORT under contract N00167-76-M-8441

UNCLASSIFIED; DISTRIBUTION UNLIMITED.

Prepared for and delivered to:

David W. Taylor Naval Ship Research and Development Center
ATTN: Code 1880, Dr. David K. Jefferson
U.S. Department of the Navy
Bethesda, MD 20084

Report Number: MISRC-TR-77-01

AN AUTOMATED ANNOTATED BIBLIOGRAPHY
ON THE SPECIFICATION OF INFORMATION SYSTEM REQUIREMENTS

Gordon C. Everest, Principal Investigator
Olin Bray
Indulis Valters

Management Information Systems Research Center
Graduate School of Business Administration
University of Minnesota
Minneapolis, MN 55455

1976 October 11

FINAL REPORT under contract N00167-76-M-8441

UNCLASSIFIED; DISTRIBUTION UNLIMITED.

Prepared for and delivered to:

David W. Taylor Naval Ship Research and Development Center
ATTN: Code 1880, Dr. David K. Jefferson
U.S. Department of the Navy
Bethesda, MD 2C084

This project was partially supported by the University of Minnesota Computer Center.

UNCLASSIFIED, DISTRIBUTION UNLIMITED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

(14) 1. REPORT NUMBER
MISRC-TR-77-Ø1

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

(9)

4. TITLE (and Subtitle)
(6) An Automated Annotated Bibliography on the Specification of Information System Requirements,

5. ~~TYPE OF REPORT & PERIOD COVERED~~
FINAL REPORT. 1 Aug-
~~1976 August~~ 11 Octo[redacted] 76,

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
(10) Gordon C./Everest, Principal Investigator
Olin/Bray
Indulis/Valters

8. CONTRACT OR GRANT NUMBER(s)
(15) NØØ167-76-M-8441

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Management Information Systems Research Center
Graduate School of Business Administration
Univ. of Minnesota, Minneapolis, MN 55455

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS
David W. Taylor Naval Ship Research & Development
Center; ATTN: Code 1880, Dr. David K. Jefferson
U.S. Department of the Navy; Bethesda, MD 20084

12. REPORT DATE
1976 October 11

13. NUMBER OF PAGES
153

14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)
(11) 11 Oct 76

15. SECURITY CLASS. (of this report)
UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)
DISTRIBUTION UNLIMITED. (12) 154p.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)
Annotated Bibliography, Information System Requirements, Problem Statement Language, Database Management Systems.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Report of an automated, annotated Bibliography containing 43 items. All relate to the area of capturing and expressing as a language, a specification of information system requirements. Each item assigned descriptors and annotated from the perspective of database technology and database management systems. Report contains a description of the data items in the bibliographic schema and output consisting of (1) the bibliographic entries with descriptors and annotations, (2) Author index, (3) Descriptor index, (4) Thesaurus of descriptors.

UNCLASSIFIED, DISTRIBUTION UNLIMITED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

409153

4B

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

Table of Contents

AN AUTOMATED ANNOTATED BIBLIOGRAPHY
ON THE SPECIFICATION OF INFORMATION SYSTEM REQUIREMENTS

Introduction and Overview

This is the final report under contract #N00167-76-M-8441 with the Naval Ship Research and Development Center, entitled "An Automated Annotated Bibliography on the Specification of Information System Requirements". It represents the initial step in a continuing effort to gather bibliographic information relating to a defined research area. Each bibliographic entry is chosen for its relevance to this research area and each is annotated and assigned descriptors from the perspective of this research area.

The research area is concerned with the capturing of information system requirements, the development of a language for expressing information system requirements, and the use of the information system requirements specification for file design and application system design procedures. The defined research area is viewed primarily from the perspective of database technology and database management systems.

The bibliography for this report contains 43 entries. Each entry along with a set of subject descriptors is recorded on a SYSTEM 2000 database. In addition, an annotation was prepared manually for each bibliographic entry.

Summary of Methodology

1. Search out and identify bibliographic items which relate to the defined research area.
2. Develop the logical schema for a bibliographic entry, develop the formal SYSTEM 2000 database definition, and design the input form for data capture.
3. For each item reviewed and found to contain some useful and relevant material,
    a. prepare the annotation for typing
    b. assign descriptors and descriptions
    c. fill out the input form for the automated bibliographic system

4. Obtain the output to prepare a final report under this contract:
   a. printout of bibliographic entries including bibliographic citation, descriptors, and incorporating the manually prepared annotation, ordered by identifier
   b. index by author
   c. tally of descriptor terms yielding the thesaurus
   d. index by descriptor

<u>Summary Schema of Bibliographic Entry</u>

The following diagram contains the essential information which makes up each bibliographic entry. A full definition of the schema is contained in Appendix I.

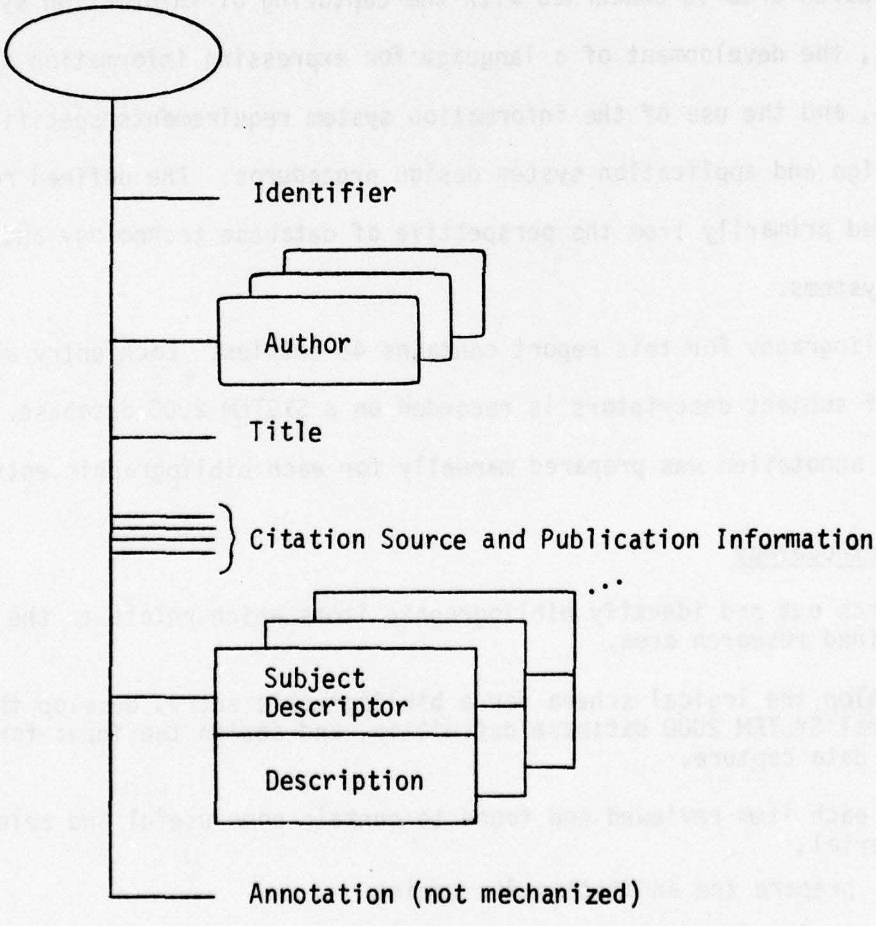

Identifier
Author
Title
Citation Source and Publication Information
...
Subject
Descriptor
Description
Annotation (not mechanized)

Figure 1. Summary Schema of Bibliographic Entry

Criteria for Inclusion of a Bibliographic Item

Bibliographic items were selected on the basis of their relevance to the defined research area. In particular, we looked for articles that were comprehensive, broad, contained something unique, or served to advance the thinking in some aspect of the subject. In some cases, items were included simply because most would agree that they should be there.

Formation of Thesaurus of Descriptors

The total set of descriptors used in preparing this automated bibliography was formed by a combination of bottom-up and top-down development. The top-down development stemmed from three considerations:

1. Being within the scope of the research area.
2. From a database management perspective.
3. Toward the goal of developing the information content and semantics of an information system requirements specification language.

The top-down development of the thesaurus of descriptors was influenced substantially by initial developments under the second research contract (#N00167-76-M-8476) for the Naval Ship Research and Development Center entitled "Developing a Data Perspective on the Specification of Information System Requirements". Some of the work for the second contract was undertaken concurrently with the work under this contract. The top-down formation of descriptors begins with data structure, patterns of processing, and behavioral characteristics. Further detail can be found in Figure 2. The terms and concepts reflected in Figure 2 are only intended to be illustrative. They are not complete. In so far as possible, most descriptors relate to this taxonomy, but not all. This taxonomy of terms evolved as the preparation of annotations for bibliographic entries progressed.

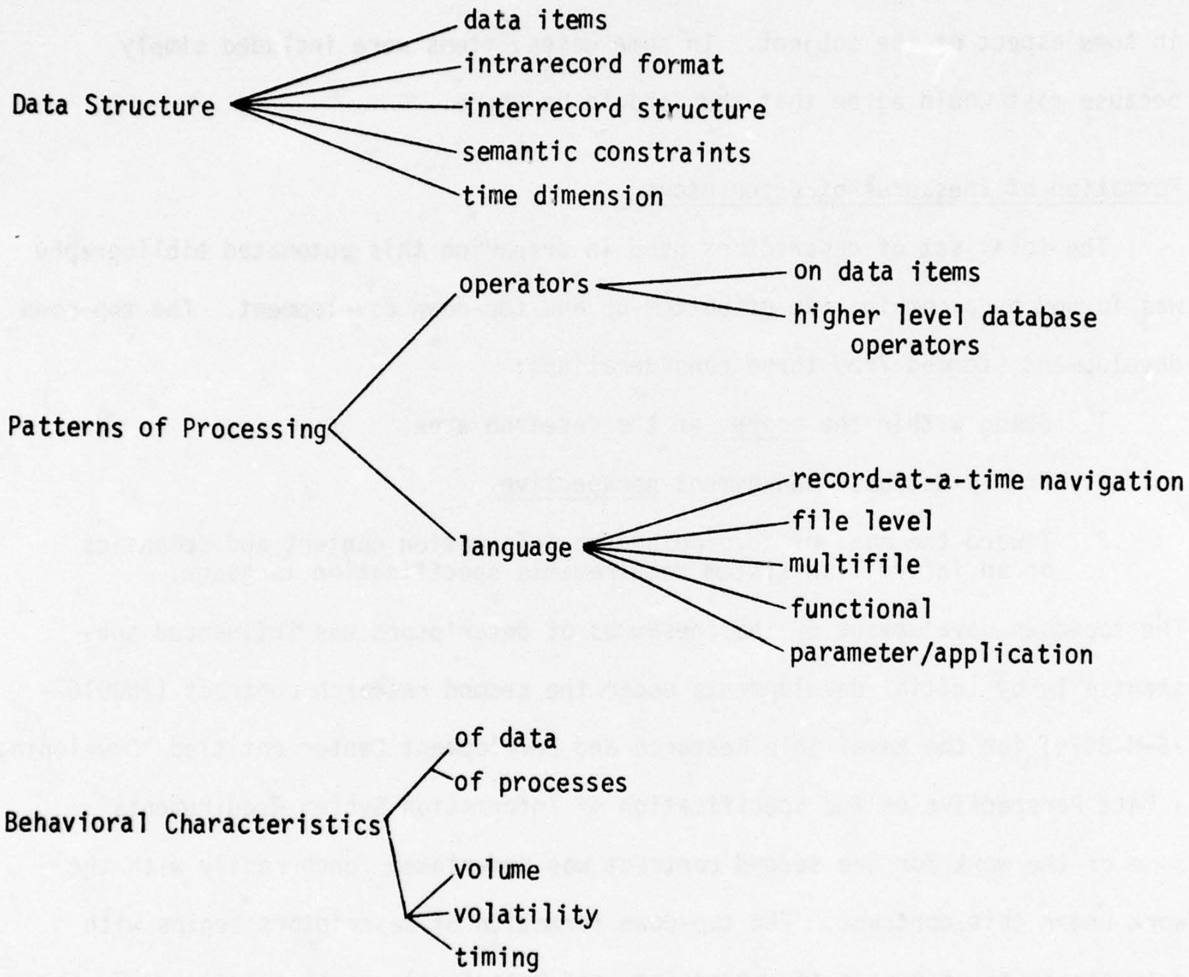


Figure 2. <u>Initial Top-Down Formation of Descriptor Thesaurus</u>

The thesaurus of descriptors was formed in a bottom-up fashion in the sense that once an initial taxonomy of terms was established, we then used whatever terms seemed most appropriate with respect to each individual bibliographic entry. No attempt was made to obtain completeness or absolute consistency. No complete thesaurus was constructed before assigning descriptors to bibliographic entries.

Preparing the Annotations

For each bibliographic entry, an annotation was prepared according to the following set of guidelines:

1. It was written as it related to the defined research area and the taxonomy of terminology formed as described in the previous section.
2. A brief summary of the bibliographic item.
3. To highlight the unique aspects of the bibliographic item.
4. To highlight the better or fuller treatments of a subject.
5. Highlight the omissions and unanswered questions.
6. Provide critique and evaluation.
7. Add our own perceptions, extensions, or generalizations.
8. Cast the annotation in our own terminology rather than that of the article, although the relationship is noted whenever it is not obvious.

Assigning Descriptors

Having written the annotation according to the previous guidelines, it was a fairly straightforward process of pulling out the descriptors for each bibliographic entry -- they were derived directly from the annotation.

Descriptions of the Descriptors

Associated with each descriptor of each bibliographic entry is a description. The description is a textual field intended to further modify or explain the descriptor as it related to that bibliographic entry. This is felt to be a unique and valuable contribution of the bibliography developed under this contract.

The scheme of modifying the descriptor aids in searching the bibliography. It enables a user of the bibliography to increase the relevance ratio. That is, having retrieved some bibliographic entries based upon a set of descriptors, the user can apply a second level of selection by reading the descriptions associated with each descriptor of each bibliographic entry.

Input and Maintenance of Bibliographic Entries

The bibliographic citation information and descriptors with associated descriptions were captured on an input data form. The input form was designed so that data could be input directly to SYSTEM 2000 in the form it was captured. The information was punched onto 80-column cards before being entered into the system. Once the database was established, it could be corrected or modified online or directly on the cards. We chose the latter and, therefore, would re-create the database each time some corrections were made to the bibliographic entries. This presents no problem for a small database and provides convenient backup protection. When the database is larger it is possible to dump the database to tape for backup.

Output

Four major output reports were obtained from the bibliographic system. These reports are contained in Appendix 2. The following is a brief description of each report:

1. Bibliographic Entries

   This report contains a printout of all the information in the bibliographic system. Each bibliographic entry is printed with the title, author(s), source, publication, descriptors and descriptions, along with the manually prepared annotation. The output is ordered according to bibliographic entry identifier. Since the identifier is formed from a combination of year published and author name, the output is first ordered by year and then ordered by author.

2. Author Index

The second report simply contains a printout of authors and for each provides the one or more bibliographic entry identifiers related to that author.

3. Tally of Descriptors -- Thesaurus

SYSTEM 2000 provides a unique feature of being able to tally all the unique values of any particular data item within the database. By doing this on the descriptor data item, we effectively obtain a printout of the thesaurus used in the bibliography. This does not represent a controlled thesaurus. Some cleanup for consistency was performed on the descriptors. Additional manipulation and improvement of the thesaurus is possible. For example, we could add cross-references within the set of descriptors (including both "conditions on query", "query conditions", say).

4. Descriptor Index

The fourth report from the bibliography consists of a printout of each descriptor followed by the relevant citations associated with that descriptor and for each citation the description associated with that descriptor. We initially attempted to print each descriptor only once, but this proved to be impossible using the PRINT facility of SYSTEM 2000. Therefore, the descriptor appears printed out before each individual citation.

Online Availability of the Bibliographic Database

The bibliographic database prepared under this contract was established and is maintained under SYSTEM 2000. The system can be accessed online from a remote terminal. Since the bibliography continues to grow and evolve, readers of this report may desire to access the latest version of the bibliography. Appropriate arrangements can be made by contacting the principal investigator:

Dr. Gordon C. Everest
Management Information Systems Research Center
Graduate School of Business Administration
UNIVERSITY OF MINNESOTA
Minneapolis, MN 55455
Phone (612) 373-5601

Conclusion and Observations

Having prepared an initial taxonomy of concepts and terminology relating to our particular perspective on the research area, the writing of annotations and assigning of descriptor terms proved to be an extremely fruitful and enlighting exercise. It provided substantial insight into the relationship between what other authors are saying about this subject area and the way we have been thinking of this subject research area. In relation to the next and following contracts in this area, this bibliography will provide substantial input and support. It will be a useful continuing process in support of future contract activities.

One significant observation is the distinction between what the user must specify regarding his information system requirements and what is typically required as input for a database management system or file design procedure or application system design procedure. Some of the required input information is exogenous and basic to the way the user thinks; other information is more properly derived from what the user would specify. For example, most systems required the user to declare which data items in a data definition should be indexed for more rapid retrieval. This decision really rests on or is derived from information about the different types of queries against the database, their relative frequency, and their required response rate. This points up the importance of gathering behavioral information with respect to the data structure and the patterns of processing against the database. However, we noticed a distinct lack of materials that related to the specification of behavioral characteristics.

APPENDIX I

Bibliographic Entry

Data Definition

Table of Contents Page

Bibliographic Entry Data Definition

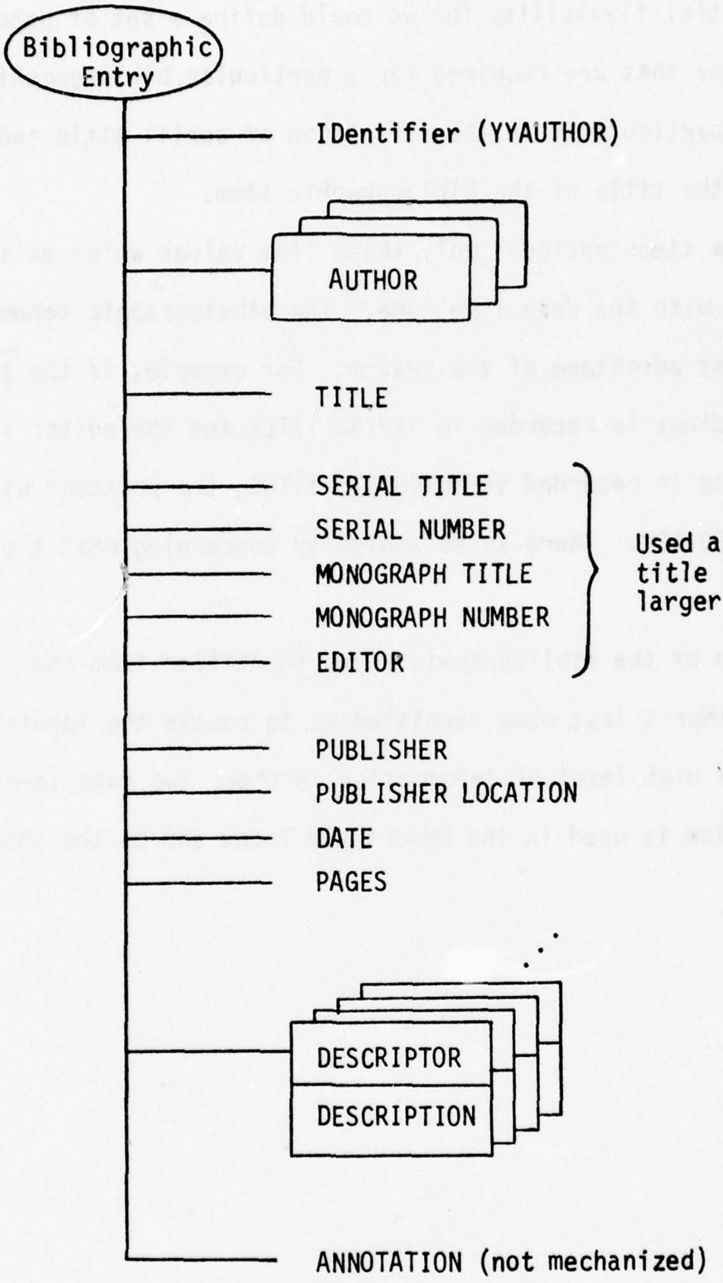
In SYSTEM 2000 all data items are considered optional by the system. This permitted substantial flexibility for we could define a set of data items and then use only those that are required for a particular bibliographic entry. This can be seen particularly in the definition of serial title and monograph title along with the title of the bibliographic item.

With all data items optional only those item values which exist will be printed out along with the data item name. The bibliographic schema was established to take best advantage of the system. For example, if the title of a conference proceedings is recorded in SERIAL TITLE and the editor is recorded in EDITOR, but nothing is recorded in MONOGRAPH TITLE, the printout will show only SERIAL TITLE and EDITOR. There is no ambiguity concerning what the editor applies to.

The formation of the bibliographic entry identifier from the year of publication and the author's last name permitted us to couple the identifier with the title and obtain a high level of information in these two data items. This two field identification is used in the Descriptor Index and on the annotations.

BIBLIOGRAPHIC ENTRY SCHEMA

GRAPHICAL REPRESENTATION



BIBLIOGRAPHY SCHEMA

DATA ITEM DESCRIPTIONS

ID YYAUTHOR: Each article is assigned a unique identifier consisting of two subfields. The first subfield is made up of the last two digits of the year the article was published; the second subfield contains the last name of the author(s). If there are two or three authors, the second subfield includes the last name of each author linked by a plus sign (first + second + third). If there are several authors, only the first author is used in the ID. If the bibliography includes more than one item for the same author in the same year, a digit is added after his name (e.g., 75SMITH2) to ensure uniqueness.

AUTHOR is a repeating group of two data items, AUTHOR NAME and FIRST NAME, with up to three instances.

AUTHOR NAME: contains only the last name of the author with no extra punctuation. If there is a corporate author, the last name is the name of the corporation. When reviewing specific Database Management Systems, the developer is listed as the author.

FIRST NAME: contains the first name of the author and any additional descriptive information about the author. This data item may contain any combination of the following:

a. author's first name and/or initials

b. additional name appendages (JR., III)

c. author role (EDITOR, COMPILER)

d. multiple author designation (ET AL.)

e. corporate author designations (INC., CO., CORP., SA, AG)

TITLE: The primary title of the bibliographic entry is always recorded in the TITLE field, whether it is the title of an article, chapter, or paper in a larger work (such as a periodical, journal, book, or conference proceedings) or it is the title of a complete work published as a self-contained volume (book, manual, report, thesis, research monograph, or working paper). If the bibliographic entry is part of a larger work, its title is enclosed in quotation marks, and the title of the larger work is given under SERIAL TITLE or MONOGRAPH TITLE. If the proceedings of a conference is published as an issue of a serial, the information is in both SERIAL TITLE and MONOGRAPH TITLE.

SERIAL TITLE: contains the title of the larger work in which the bibliographic entry TITLE appeared, but only when the larger work is published on a repeating basis with the same title (e.g., journal, periodical, conference proceedings). Below are examples of several of the more frequently referenced conferences:

PROCEEDINGS ACM NATIONAL CONFERENCE

PROCEEDINGS NATIONAL COMPUTER CONFERENCE

PROCEEDINGS ACM-SIGMOD WORKSHOP

SERIAL NUMBER: If the source is a serial, this will identify the volume and number. The format is "volume:number." If there is only a volume designation, the format "VOLUME 45" is used. The number of issue may be either an integer (15:6) or a season (15:FALL). The actual date of the serial is given below under year and date. In general, the SERIAL NUMBER field contains any necessary modifiers to the SERIAL TITLE.

MONOGRAPH TITLE: If the bibliographic entry is part of a larger self-contained volume, this field contains the title of the larger work.

MONOGRAPH NUMBER: This field provides additional information which modifies or completes the MONOGRAPH TITLE. This field contains such information as edition number (if greater than the first), NTIS number, Government report number, or number in a working paper series.

EDITOR: Identifies the editor of either a SERIAL TITLE or a MONOGRAPH TITLE. It is shown in the form "LAST NAME, FIRST NAME."

PUBLISHER: Identifies the name of the publisher who produced the self-contained volume of the bibliographic entry TITLE or the SERIAL or MONOGRAPH in which it appeared. For joint publishers, only the primary publisher is recorded. This field is often not necessary for well known SERIAL TITLES.

PUBLISHER LOCATION: City (and state) of the publisher, that is, the source for obtaining the TITLE. For proceedings it is the location of the publisher, not the conference.

YEAR: Contains the year of publication (or production if unpublished item). This field always contains a value as a four digit number.

DATE: This field is used when more information than year is relevant for the publication date. It may contain year month day as YYYY/MM/DD or year and season.

PAGES: The pages on which the article or chapter appears if it is part of a larger work (format NN-MM) or the total number of pages if the bibliographic entry is a self-contained volume (format NNN).

SUBJECTS: A repeating group containing a DESCRIPTOR and associated DESCRIPTION data items.

DESCRIPTOR: A subject term or phrase which describes the bibliographic entry as it relates to an interest in the research area. Every attempt was made to use descriptors in a way consistent with the research area and consistent across all bibliographic entries. A listing of all descriptors used so far is attached. This list is not intended to be exhaustive of the research area; it was built up by tallying the terms used with each entry. DESCRIPTORS were not assigned to be descriptive of the entire bibliographic entry; but only to capture the unique aspects of the entry as it relates to the research area.

DESCRIPTION: A text field to accompany each DESCRIPTOR for each bibliographic entry. It is used to indicate such things as:

- the viewpoint taken on the DESCRIPTOR
- more detail from the entry as it relates to the DESCRIPTOR
- the page numbers in the entry to which the DESCRIPTOR applies (if the entry is a large work)

The description field is not always used.

BIBLIOGRAPHY SCHEMA

DATA ITEM AND REPEATING GROUP DEFINITION TABLE

ITEM NAME	KEYED	FORMAT TYPE	EXISTENCE	SAMPLE VALUES
ID YYAUTHOR	KEY	YYAAAA...[n]	MANDATORY, UNIQUE	68SMITH 75JONES1 76SMITH + JONES
AUTHOR (RG--up to three instances)				
AUTHOR NAME	KEY	ALPHA ONLY (no punctuation)	MANDATORY (one).	EVEREST (Last name only)
FIRST NAME	no	Char String		GORDON ‖ INC. G.C., JR. ‖ ET AL. EDITOR ‖ CORP
TITLE	KEY	text	MANDATORY	(in quotes if part of larger work.)
SERIAL TITLE	no	text		
SERIAL NUMBER	no	Char String		15:6 15:FALL
MONOGRAPH TITLE	no	text		
MONOGRAPH NUMBER	no	Char String		THIRD EDITION AD 942 371
EDITOR	no	Char String		JONES, G.C. JR
PUBLISHER	KEY	Alpha		MCGRAW-HILL
PUBLISHER LOCATION	no	Char String		ST PAUL, MN
YEAR	KEY	NNNN	MANDATORY	1975
DATE	no	Alpha/numeric	only if more than year	1975/10 1976 FALL
PAGES	no	NN NN-MM		395 (total pages) 22-35 (from-to)
SUBJECTS (RG)				
DESCRIPTOR	KEY	alpha	MANDATORY (one instance)	
DESCRIPTOR	no	text	(one for each descriptor; optional).	
ANNOTATION	--	text	MANDATORY	(not MECHANIZED).

Bibliographic Entry Schema
Linear Representation
(SYSTEM 2000 Data Definition).

```
  1* ID YYAUTHOR-           ( KEY NAME ):
  2* TITLE-------           ( KEY NAME):
  4* SERIAL TITLE-------    ( NON-KEY NAME ):
 41* SERIAL NUMBER------    ( NON-KEY NAME ):
  5* MONOGRAPH TITLE----    ( NON-KEY NAME ):
 51* MONOGRAPH NUMBER---    ( NON-KEY NAME ):
 52* EDITOR-------------    ( NON-KEY NAME ):
  6* PUBLISHER----------    ( KEY NAME ):
 61* PUBLISHER LOCATION-    ( NON-KEY NAME ):
  7* YEAR---------------    ( KEY INT 9(4) ):
 71* DATE---------------    ( NON-KEY NAME ):
 72* PAGES--------------    ( NON-KEY NAME ):

 80* AUTHOR                 ( RG ):
    8* AUTHOR NAME-         ( KEY NAME IN 80 ):
   81* FIRST NAME--         ( NON-KEY NAME IN 80 ):

 90* SUBJECTS               ( RG ):
    9* DESCRIPTOR--         ( KEY NAME IN 90 ):
   91* DESCRIPTION-         ( NON-KEY NAME IN 90 ):
```

Naval Ship R&D Center
BIBLIOGRAPHY INPUT FORM

ID YYAUTHOR: 1* *

TITLE: 2*

*

SERIAL TITLE 4* *

SERIAL NUMBER: 41* *

MONOGRAPH TITLE: 5*

*

MONOGRAPH NUMBER: 51* *

EDITOR: 52* *

PUBLISHER: 6* *

PUBLISHER LOCATION: 61* *

YEAR: 7* *

DATE: 71* *

PAGES: 72* *

AUTHOR NAME: 80*8* *

FIRST NAME: 81* *

AUTHOR NAME: 80*8* *

FIRST NAME: 81* *

AUTHOR NAME: 80*8* *

FIRST NAME: 81* *

DESCRIPTOR: 90*9* *

DESCRIPTION: 91*

*

DESCRIPTOR: 90*9* *

DESCRIPTION: 91*

*

DESCRIPTOR: 90*9* *

DESCRIPTION: 91*

* □
OVER

DESCRIPTOR: 90*9*______________________________*

DESCRIPTION: 91*______________________________

____

______________________________*

DESCRIPTOR: 90*9*______________________________*

DESCRIPTION: 91*______________________________

____

______________________________*

DESCRIPTOR: 90*9*______________________________*

DESCRIPTION: 91*______________________________

____

______________________________*

DESCRIPTOR: 90*9*______________________________*

DESCRIPTION: 91*______________________________

____

______________________________*

DESCRIPTOR: 90*9*______________________________*

DESCRIPTION: 91*______________________________

____

______________________________*

DESCRIPTOR: 90*9*______________________________*

DESCRIPTION: 91*______________________________

____

______________________________*

DESCRIPTOR: 90*9*______________________________*

DESCRIPTION: 91*______________________________

____

______________________________*

DESCRIPTOR: 90*9*______________________________*

DESCRIPTION: 91*______________________________

____

______________________________*

APPENDIX II

Output Reports

Table of Contents

1. TALLY OF DESCRIPTORS -- THESAURUS

```
*************************
 ELEMENT-      DESCRIPTOR--
*************************
FREQUENCY    VALUE
-------------------------
        2        ABSTRACT DATA TYPE
        8        ACCESS CONTROL
        2        ACCESS CONTROL MECHANISMS
        3        ACCESS PATH
        1        ACCESS PRIVILEGES
        1        APPLICATION SYSTEM
        1        ATTRIBUTES
        1        AUDIT
        1        AUDITOR
        3        BACKUP AND RECOVERY
        1        BACKUP DATA
        1        BACKUP STRATEGY
        3        BEHAVIORAL CHARACTERISTICS
        1        COMPOUND DATA STRUCTURE OPERATORS
        1        COMPRESSION OPERATOR
        1        CONCURRENCY USER COMMANDS
        2        CONCURRENT CONTROL
        1        CONCURRENT PROCESSES
        3        CONDITIONS ON A PROCESS
        1        CREATION
        1        DATA CONSISTENCY RULES
        8        DATA DEFINITION
        2        DATA DEPENDENT ACCESS CONTROL
        1        DATA DEPENDENT UPDATE
        1        DATA DICTIONARY
        3        DATA INTEGRITY
        6        DATA ITEM DEFINITION
        1        DATA ITEM DERIVATION RULES
        1        DATA ITEM OPERATORS
        2        DATA LOSS
        4        DATA STRUCTURE
        1        DATA STRUCTURE DEFINITION
        1        DATA STRUCTURE MODEL
        1        DATA STRUCTURE OPERATORS
        2        DATABASE DESIGN
        1        DATABASE DESIGN PROCEDURE
        9        DATABASE MANAGEMENT SYSTEM
        1        DATABASE OPERATIONS
        7        DATABASE OPERATORS
        1        DATABASE SYSTEM DESIGN OBJECTIVES
        1        CATE SENSITIVE FILE
        1        DEADLOCK
        1        DESIGN DATABASE
        1        DESIGN TRADEOFFS
        1        DEVICE INDEPENDENCE
        4        ENCODE/DECODE
        2        ENCRYPTION
        1        EVOLVABILITY
        1        EXCEPTION CONDITION ACTIONS
        2        EXCEPTION CONDITION RESPONSES
        2        EXCEPTION CONDITIONS
        1        EXCEPTION PROCESSING
        2        FEATURE LIST
        1        FILE LEVEL DERIVATION
        1        FLAT FILE
        1        FLOW DATA
        1        FORMATION AND PRESENTATION
        1        FUNCTIONS
        1        GRAPHICAL OUTPUT
        1        HIERARCHICAL SELECTION CRITERIA
        1        HIGH LEVEL LANGUAGE
        4        HIGH LEVEL MULTIFILE LANGUAGE
        1        INDEX DEFINITION
        1        INFERENTIAL RETRIEVAL
        3        INFORMATION ALGEBRA
        1        INFORMATION REQUIREMENTS
        1        INFORMATION SYSTEM MODELLING
        2        INFORMATION SYSTEM REQUIREMENTS SPECIFICATIONS
        1        INFORMATION SYSTEM REQUIREMENTS STATEMENT LANGUAGE
        1        INTERROGATION
        1        LOCKS ON DATA
        1        LOGICAL PAGE SIZE
        1        OBJECTIVES
        3        PATTERNS OF PROCESSING
        1        PERFORMANCE CRITERIA
        2        PERFORMANCE MONITORING
        2        PREMISE ACTION
        3        PROCESS DEFINITION
        1        PROGRAM GENERATION
        1        PSL
        1        QUERY CONDITIONS
        1        QUERY RESPONSES
        2        QUERY TYPES
        1        RECOVERY
        1        RECOVERY PROCEDURES
        7        REPORT DEFINITION
        2        REPORT GENERATION
        1        RESTART
        1        RETENTION
        1        RULES OF INFERENCE
        1        SECURITY
        1        SECURITY MONITORING
        4        SELECTION CRITERIA
        2        SEMANTIC CONSTRAINTS
        1        SEMANTIC INTEGRITY
        1        STATUS DATA
        1        STRING ITEM TYPE
        2        SURVEY OF DMS
        1        SYSTEM DEVELOPMENT PHASES
        1        TEXT ITEM TYPE
        1        TIME
        2        TIME STAMPING
        1        TRADEOFFS
        3        TRANSACTION DEFINITION
        2        TRANSACTION PROCESSING
        1        TRANSACTION VALIDATION CRITERIA
        1        TRIGGER
        2        UPDATE
        3        UPDATE PROCESSES
        3        USERSCHEMA
        1        VALIDATION
        6        VALIDATION CRITERIA
        1        VALIDATION PROCESS
        2        VALIDATION RESPONSE
---------------------
     114  UNIQUE VALUES
---------------------
     213  OCCURRENCES
---------------------
```

2. AUTHOR INDEX

```
  AUTHOR NAME                            ID YYAUTHOR
***
* ASTRAHAN                               76ASTRAHAN
* BLEIER                                 70BLEIER
* BURROUGHS                              75BURROUGHS
* CHAMBERLIN                             75ESWARAN + CHAMBERLIN
* CODASYL SYSTEMS COMMITTEE              71CODASYL
* COLLMEYER                              73KING + COLLMEYER
* COM SHARE                              73COMSHARE
* COMPUTER ASSOCIATES                    75COMPUTER ASSOCIATES
* COMPUTER CORPORATION OF AMERICA        74COMPUTER CORPORATION OF AMERICA
* COMPUVISOR                             71COMPUVISOR
* CONTROL DATA                           70CDC
* CONWAY                                 71CONWAY + MAXWELL + MORGAN
* DANA                                   72DANA + PRESSER
* DIJKSTRA                               76DIJKSTRA
* ESWARAN                                75ESWARAN + CHAMBERLIN
* FOSTER                                 73FOSTER
* GOODENOUGH                             75GOODENOUGH
* GRIFFITHS                              76GRIFFITHS + WADE
* HAMMER                                 74HAMMER
* HAMMER                                 75HAMMER + MCLEOD
* HOFFMAN                                71HOFFMAN
* IBM                                    70IBM
* KAHN                                   76KAHN
* KING                                   73KING + COLLMEYER
* LEAVENWORTH                            74LEAVENWORTH + SAMMET
* LINDGREEN                              74LINDGREEN
* LISKOV                                 74LISKOV + ZILLES
* LYNCH                                  69LYNCH
* MAXWELL                                71CONWAY + MAXWELL + MORGAN
* MCLEOD                                 75HAMMER + MCLEOD
* MINSKY                                 74MINSKY
* MORGAN                                 71CONWAY + MAXWELL + MORGAN
* NUNAMAKER                              73NUNAMAKER + SWENSON + WHINSTON
* OLLE                                   74OLLE
* PETERSON                               74SNUGGS + POPEK + PETERSON
* POPEK                                  74SNUGGS + POPEK + PETERSON
* PRESSER                                72DANA + PRESSER
* PROGRAM PRODUCTS                       74PROGRAM PRODUCTS
* RANDALL                                74RANDALL
* SAMMET                                 74LEAVENWORTH + SAMMET
* SAYANI                                 74SAYANI
* SCHEUERMANN                            74SHNEIDERMAN + SCHEUERMANN
* SENKO                                  72SENKO
* SHNEIDERMAN                            74SHNEIDERMAN + SCHEUERMANN
* SKINNER                                74SKINNER
* SNUGGS                                 74SNUGGS + POPEK + PETERSON
* SWENSON                                73NUNAMAKER + SWENSON + WHINSTON
* TAGGART                                74TAGGART
* TEICHROEW                              72TEICHROEW
* TONIK                                  71TONIK
* TSICHRITZIS                            75TSICHRITZIS
* UNIVAC                                 75UNIVAC
* U.S. DEPARTMENT OF DEFENSE             67U.S. DOD
* WADE                                   76GRIFFITHS + WADE
* WEBER                                  75WEBER
* WHINSTON                               73NUNAMAKER + SWENSON + WHINSTON
* ZILLES                                 74LISKOV + ZILLES
```

3. DESCRIPTOR INDEX

```
  ABSTRACT DATA TYPE
74HAMMER
*DATA ABSTRACTIONS FOR DATA BASES*
  DATA STRUCTURE PLUS ALLOWED OPERATORS

  ABSTRACT DATA TYPE
74LISKOV + ZILLES
*PROGRAMMING WITH ABSTRACT DATA TYPES*
  LOGICAL DATA TYPE OR STRUCTURE CHARACTERIZED BY THE OPE
       RATIONS AVAILABLE ON IT

  ACCESS CONTROL
74MINSKY
*ON INTERACTION WITH DATA BASES*
  DATA DEPENDENT

  ACCESS CONTROL
71CODASYL
FEATURE ANALYSIS OF GENERALIZED DATA BASE MANAGEMENT SYST
    EMS
  PRIVACY LOCKS BY DATA ELEMENT (AT ANY LEVEL) OR BY FUNC
       TION

  ACCESS CONTROL
76GRIFFITHS + WADE
*AN AUTHORIZATION MECHANISM FOR A RELATIONAL DATABASE SYS
    TEM*
  GRANTING AND REVOKING ACCESS PRIVILEGES. FORMING NETWOR
       K OF PRIVILEGES.

  ACCESS CONTROL
70CDC
MARS III REFERENCE MANUAL
  PRIVACY LEVELS

  ACCESS CONTROL
75ESWARAN + CHAMBERLIN
*FUNCTIONAL SPECIFICATIONS OF A SUBSYSTEM FOR DATA BASE I
    NTEGRITY*
  BY FUNCTION, ESPECIALLY FOR MODIFYING VALIDATION CRITER
       IA

  ACCESS CONTROL
70IBM
GIS VERSION 2 APPLICATION DESCRIPTION
  FILE AND ITEM LEVEL. ITEM SECURITY CODES ATTACHED TO EA
       CH USER PASSWORD

  ACCESS CONTROL
76ASTRAHAN
*SYSTEM R: RELATIONAL APPROACH TO DATABASE MANAGEMENT*
  USER GRANTS AUTHORIZATION FOR READ, CHANGE, INSERT, DEL
       ETE, CREATE LINKS OR VIEWS

  ACCESS CONTROL
71HOFFMAN
*THE FORMULARY MODEL FOR FLEXIBLE PRIVACY AND ACCESS CONT
    ROLS*
  USER WRITTEN AUTHORIZATION ROUTINES WHICH EXECUTE AT RU
       N TIME AND CAN USE DATA DEPENDENT CRITERIA

  ACCESS CONTROL MECHANISMS
```

```
71CONWAY + MAXWELL + MORGAN
*ON THE IMPLEMENTATION OF SECURITY MEASURES IN INFORMATIC
     N SYSTEMS*
   MATRIX, ROWS ARE USERS, COLUMNS ARE DATA ITEMS, ELEMENT
        IS AUTHORIZATION ACTION

   ACCESS CONTROL MECHANISMS
71COMPUVISOR
ASAP FILE MAINTENANCE AND INFORMATION RETRIEVAL SYSTEM RE
     FERENCE MANUAL
   CONTROLS ACCESS TO DEFINED ITEMS, SYSTEM FUNCTIONS, DEF
        INED PROCESSES, DEFINENEW ITEMS OR RECORDS, DATA DEPENDENT CRIT
        ERIA

   ACCESS PATH
75BURROUGHS
DMSII DATA AND STRUCTURE DEFINITION LANGUAGE(DASDL)REFERE
     NCE MANUAL
   DIRECT, CALC, VIA

   ACCESS PATH
75TSICHRITZIS
LSL: A LINK AND SELECTOR LANGUAGE
   DYNAMICALLY CREATED, MAINTAINED, AND DESTROYED

   ACCESS PATH
76ASTRAHAN
*SYSTEM R: RELATIONAL APPROACH TO DATABASE MANAGEMENT*
   DEFINITION OF INDEXES(IMAGES)AND INTER-ENTITY RELATIONS
        HIPS(LINKS)

   ACCESS PRIVILEGES
76GRIFFITHS + WADE
*AN AUTHORIZATION MECHANISM FOR A RELATIONAL DATABASE SYS
     TEM*
   FOR READ, INSERT, DELETE, UPDATE, OR DROP(ALSO IN TERMS
        OF ALL OF ALL BUT)

   APPLICATION SYSTEM
73NUNAMAKER + SWENSON + WHINSTON
*SPECIFICATIONS FOR THE DEVELOPMENT OF A GENERALIZED DATA
     BASE PLANNING SYSTEM*
   AS IT RELATES TO DMS

   ATTRIBUTES
74LINDGREEN
*BASIC OPERATIONS ON INFORMATION AS A BASIS FOR DATA BASE
     DESIGN*
   MAY DESCRIBE, IDENTIFY, OR RELATE ENTITIES. DOMAIN OF V
        ALUES FOR.

   AUDIT
74COMPUTER CORPORATION OF AMERICA
CCA204 USER LANGUAGE REFERENCE MANUAL
   PRINTS USER SPECIFIED AUDIT TRAIL

   AUDITOR
75WEBER
*AUDIT CAPABILITIES OF SOME DATABASE MANAGEMENT SYSTEMS*
   ROLE IN DMS. USE OR NON-USE OF DMS FOR AUDIT FUNCTION

   BACKUP AND RECOVERY
```

```
74SAYANI
#RESTART AND RECOVERY IN A TRANSACTION-ORIENTED INFORMATI
      ON PROCESSING SYSTEM#
   CLASSIFIES TYPES OF DATA LOSS

   BACKUP AND RECOVERY
75BURROUGHS
DMSII DATA AND STRUCTURE DEFINITION LANGUAGE(DASDL)REFERE
      NCE MANUAL

   BACKUP AND RECOVERY
71TONIK
#RECOVERY OF ONLINE DATA BASES#
   THREE CASE STUDIES, TYPES OF ERRORS, BACKUP STRATEGIES,
         AND RECOVERY PROCEDURES

   BACKUP DATA
74RANDALL
#INTEROGATING DATE-SENSITIVE FILES#
   USED TO KEEP DATA FOR PAST TIME PERIODS

   BACKUP STRATEGY
74SAYANI
#RESTART AND RECOVERY IN A TRANSACTION-ORIENTED INFORMATI
      ON PROCESSING SYSTEM#

   BEHAVIORAL CHARACTERISTICS
71TONIK
#RECOVERY OF ONLINE DATA BASES#
   OF DATA LOSSES AND SYSTEM FAILURES

   BEHAVIORAL CHARACTERISTICS
76KAHN
#A METHOD FOR DESCRIBING INFORMATION REQUIRED BY THE DATA
      BASE DESIGN PROCESS#
   DATA VOLUME, ACCESS AND PERFORMANCE STATISTICS, INTEGRI
        TY CRITERIA, ACCESS CONTROL, PROCESS PRECEDENCE ORDERING, USE F
        REQUENCY, TIMING

   BEHAVIORAL CHARACTERISTICS
67U.S. DOD
#OPERATIONAL CAPABILITY DESCRIPTION FOR A DATA MANAGEMENT
       SYSTEM#
   OF FILES, DYNAMICS, GROWTH, CHANGE

   COMPOUND DATA STRUCTURE OPERATORS
74SHNEIDERMAN + SCHEUERMANN
#STRUCTURED DATA STRUCTURES#
   BUILT UP FROM PRIMITIVE OPERATORS

   COMPRESSION OPERATOR
71COMPUVISOR
ASAP FILE MAINTENANCE AND INFORMATION RETRIEVAL SYSTEM RE
      FERENCE MANUAL
   ON TEXT ITEM TYPES

   CONCURRENCY USER COMMANDS
73KING + COLLMEYER
#DATABASE SHARING -- AN EFFICIENT MECHANISM FOR SUPPORTIN
      G CONCURRENT PROCESSES
   LOCK/UNLOCK PROCESS, ALLOCATE/DEALLOCATE DATA
```

```
  CONCURRENT CONTROL
74COMPUTER CORPORATION OF AMERICA
CCA204 USER LANGUAGE REFERENCE MANUAL
  LOCKS AT FILE LEVEL FOR MORE THAN ONE UPDATE

  CONCURRENT CONTROL
76ASTRAHAN
≠SYSTEM R: RELATIONAL APPROACH TO DATABASE MANAGEMENT≠

  CONCURRENT PROCESSES
73KING + COLLMEYER
≠DATABASE SHARING -- AN EFFICIENT MECHANISM FOR SUPPORTIN
     G CONCURRENT PROCESSES

  CONDITIONS ON A PROCESS
75ESWARAN + CHAMBERLIN
≠FUNCTIONAL SPECIFICATIONS OF A SUBSYSTEM FOR DATA BASE I
     NTEGRITY≠
  FOUR CLASSIFICATIONS-(1)ITEM DOMAIN VS INTER-ITEM CONSI
         STENCY, (2)STATE TRANSITION VS TRANSACTION(3)IMMEDIATE(ALWAYS H
         OLDS) VS DELAYED(SPECIFY WHEN TO TEST),AND(4)INVOKED ON DEMAND

  CONDITIONS ON A PROCESS
76DIJKSTRA
A DISCIPLINE OF PROGRAMMING

  CONDITIONS ON A PROCESS
74OLLE
≠SELF CONTAINED FACILITIES TO PROCESS SUB STRUCTURES OF M
     ORE COMPLEX NETWORK STRUCTURES≠
  PREMISE ACTION SEPCIFICATION

  CREATION
73COMSHARE
COMPOSIT 77 USERS GUIDE
  RULES FOR HANDLING MISSING DATA

  DATA CONSISTENCY RULES
74MINSKY
≠ON INTERACTION WITH DATA BASES≠

  DATA DEFINITION
75ESWARAN + CHAMBERLIN
≠FUNCTIONAL SPECIFICATIONS OF A SUBSYSTEM FOR DATA BASE I
     NTEGRITY≠
  PROPOSES DYNAMIC MODIFICATION OF VALIDATION CRITERIA

  DATA DEFINITION
74PROGRAM PRODUCTS
THE DATA ANALYZER TECHNICAL REFERENCE MANUAL
  MAINLY PHYSICAL(DISPLACEMENT, LENGTH, TYPE). TEMPORARY
         VARIABLES FOR CALCULATION OR OUTPUT

  DATA DEFINITION
69LYNCH
≠AUS:A TECHNIQUE IN SYSTEM DOCUMENTATION≠
  FLAT FILES AND COORDINATED FILES

  DATA DEFINITION
75BURROUGHS
DMSII DATA AND STRUCTURE DEFINITION LANGUAGE(DASDL)REFERE
     NCE MANUAL
  LOGICAL AND PHYSICAL STRUCTURE AND VALIDATION CRITERIA
       FOR INCLUSION OF RECORD IN THE FILE

  DATA DEFINITION
70IBM
GIS VERSION 2 APPLICATION DESCRIPTION
  INCLUDES DEFAULTS ON OUTPUT---COLUMN HEADING, EDITING,
       AND ENCODE-DECODE TABLE

  DATA DEFINITION
67U.S. DOD
≠OPERATIONAL CAPABILITY DESCRIPTION FOR A DATA MANAGEMENT
     SYSTEM≠
  INCLUDING OPERATIONS

  DATA DEFINITION
75TSICHRITZIS
LSL: A LINK AND SELECTOR LANGUAGE
  SCHEMA, SUBSCHEMA, VALIDITY CRITERIA, AND ACCESS PATHS

  DATA DEFINITION
70CDC
MARS III REFERENCE MANUAL

  DATA DEPENDENT ACCESS CONTROL
71CONWAY + MAXWELL + MORGAN
≠ON THE IMPLEMENTATION OF SECURITY MEASURES IN INFORMATIO
    N SYSTEMS≠
  MUST BE CHECKED AT RUN TIME

  DATA DEPENDENT ACCESS CONTROL
74MINSKY
≠ON INTERACTION WITH DATA BASES≠
  AT INSTANCE LEVEL; SIMILAR TO CLASS IN SIMULA

  DATA DEPENDENT UPDATE
73KING + COLLMEYER
≠DATABASE SHARING -- AN EFFICIENT MECHANISM FOR SUPPORTIN
    G CONCURRENT PROCESSES
  USER RESPONSIBILITY IN CONCURRENT ACCESS

  DATA DICTIONARY
74TAGGART
≠DEVELOPING AN ORGANIZATION≠S INFORMATION INVENTORY≠
  DEFINITION OF INFORMATION ELEMENTS, USE OF DATA DICTION
       ARY

  DATA INTEGRITY
74SNUGGS + POPEK + PETERSON
≠DATA BASE SYSTEM OBJECTIVES AS DESIGN CONSTRAINTS≠
  BACKUP AND RECOVERY, CONCURRENCY CONTROL, ACCESS CONTRO
       L, ENCRYPTION

  DATA INTEGRITY
76ASTRAHAN
≠SYSTEM R: RELATIONAL APPROACH TO DATABASE MANAGEMENT≠
  MAINTAINED BY ASSERTIONS WHICH DESCRIBE PERMISSIBLE STA
       TES OF THE DATABASE OR PERMISSIBLE TRANSITIONS OF DATABASE

  DATA INTEGRITY
75ESWARAN + CHAMBERLIN
≠FUNCTIONAL SPECIFICATIONS OF A SUBSYSTEM FOR DATA BASE I
```

BEST AVAILABLE COPY

```
    NTEGRITY≠
  FUNCTIONS INCLUDE ACCESS CONTROL, CONCURRENT UPDATE CON
        TROL, BACKUP AND RECOVERY, AND VALIDATION

  DATA ITEM DEFINITION
74TAGGART
≠DEVELOPING AN ORGANIZATION≠S INFORMATION INVENTORY≠
  ATTRIBUTE, TIME, ENTITY DESCRIBED

  DATA ITEM DEFINITION
75BURROUGHS
DMSII DATA AND STRUCTURE DEFINITION LANGUAGE(DASDL)REFERE
     NCE MANUAL
  NAME, TYPE, SIZE, INITIAL AND NULL VALUES, NUMBER OF OC
        CURRENCES IF REPEATING, AND MANDATORY OR OPTIONAL

  DATA ITEM DEFINITION
76KAHN
≠A METHOD FOR DESCRIBING INFORMATION REQUIRED BY THE DATA
     BASE DESIGN PROCESS≠
  VALUE SET DISTRIBUTION

  DATA ITEM DEFINITION
73COMSHARE
COMPOSIT 77 USERS GUIDE
  ENCODE-DECODE AND ENCRYPTION

  DATA ITEM DEFINITION
70BLEIER
≠DATA DEFINITION STANDARDIZATION≠
  LIST 20 ATTRIBUTES, MOSTLY PHYSICAL, INCLUDING NAME, SI
        ZE, CODING TYPE, KEY, EDITING, ENCODING, DECODING, ACCESS CONTR
        OLS, AND VALIDATION CRITERIA

  DATA ITEM DEFINITION
71CODASYL
FEATURE ANALYSIS OF GENERALIZED DATA BASE MANAGEMENT SYST
     EMS
  NAME, LENGTH, VALUE SET, UNITS OR DIMENSIONS, EXISTENC
        E, ACCESS LOCKS, USAGE, ACTUAL OR DERIVED, SINGLE OR MULTIVALUE
        D, AND EDITING RULES

  DATA ITEM DERIVATION RULES
69LYNCH
≠ADS:A TECHNIQUE IN SYSTEM DOCUMENTATION≠

  DATA ITEM OPERATORS
75ESWARAN + CHAMBERLIN
≠FUNCTIONAL SPECIFICATIONS OF A SUBSYSTEM FOR DATA BASE I
     NTEGRITY≠
  DEFINE ITEM TYPES WHICH CAN BE VALID OPERANDS

  DATA LOSS
71TONIK
≠RECOVERY OF ONLINE DATA BASES≠
  TYPES OF FAILURES AND EXAMPLES OF BEHAVIORAL DATA ON TH
        OSE FAILURES

  DATA LOSS
74SAYANI
≠RESTART AND RECOVERY IN A TRANSACTION-ORIENTED INFORMATI
     ON PROCESSING SYSTEM≠
  CAUSES: HARDWARE (DATABASE CRASH), SOFTWARE, USER UPDAT
       E PROCESS

  DATA STRUCTURE
74COMPUTER CORPORATION OF AMERICA
CCA204 USER LANGUAGE REFERENCE MANUAL
  SINGLE FLAT FILE(INVERTED), LATER RELEASE FOR COORDINAT
       ED FILES. NOW MULTIPLE FLAT FILES ONE AT A TIME

  DATA STRUCTURE
73COMSHARE
COMPOSIT 77 USERS GUIDE
  FLAT FILE(SINGLE OR COORDINATED)

  DATA STRUCTURE
74LINDGREEN
*BASIC OPERATIONS ON INFORMATION AS A BASIS FOR DATA BASE
      DESIGN*
  FLAT FILE(SINGLE OR MULTIPLE)

  DATA STRUCTURE
74PROGRAM PRODUCTS
THE DATA ANALYZER TECHNICAL REFERENCE MANUAL
  SINGLE OR COORDINATED FLAT FILES(LIMITED USE OF REPEATI
       NG GROUPS --ONLY FOR SELECTION)

  DATA STRUCTURE DEFINITION
74SHNEIDERMAN + SCHEUERMANN
*STRUCTURED DATA STRUCTURES*
  DEFINED ALONG WITH THE SET OF MEANINGFUL OPERATORS ON T
       HE STRUCTURE

  DATA STRUCTURE MODEL
67U.S. DOD
*OPERATIONAL CAPABILITY DESCRIPTION FOR A DATA MANAGEMENT
      SYSTEM*
  (1)SINGLE PATH HIERARCHY,(2)MULTI-PATH HIERARCHY,(3)INV
       ERTED HIERARCHY (MULTIPLE PARENTS),(4)LINKED HIERARCHY(NETWORK)
       ,(5)LINKED LIST(POINTERS ON INSTANCES), AND(6)ASSOCIATIVE LINK(
       BY CONTENT)

  DATA STRUCTURE OPERATORS
74SHNEIDERMAN + SCHEUERMANN
*STRUCTURED DATA STRUCTURES*
  FOCUS ON PROGRAMMER, RECORD AT A TIME, NAVIGATIONAL AID
       S

  DATABASE DESIGN
76KAHN
*A METHOD FOR DESCRIBING INFORMATION REQUIRED BY THE DATA
     BASE DESIGN PROCESS*
  INFORMATION REQUIREMENTS

  DATABASE DESIGN
74LINDGREEN
*BASIC OPERATIONS ON INFORMATION AS A BASIS FOR DATA BASE
      DESIGN*
  DECIDING DATA TO BE INCLUDED AND OPERATIONS TO PERFORM
       ON DATA

  DATABASE DESIGN PROCEDURE
72SENKO
```

```
#DETAILS OF A SCIENTIFIC APPROACH TO INFORMATION SYSTEMS#
  FOREM AND PHASE II

  DATABASE MANAGEMENT SYSTEM
75COMPUTER ASSOCIATES
EARL GENERAL INFORMATION MANUAL

  DATABASE MANAGEMENT SYSTEM
73COMSHARE
COMPOSIT 77 USERS GUIDE
  ONLY ACCESSED VIA ONLINE, TIMESHARING FACILITY

  DATABASE MANAGEMENT SYSTEM
74PROGRAM PRODUCTS
THE DATA ANALYZER TECHNICAL REFERENCE MANUAL

  DATABASE MANAGEMENT SYSTEM
76ASTRAHAN
#SYSTEM R: RELATIONAL APPROACH TO DATABASE MANAGEMENT#
  IBM RESEARCH SYSTEM

  DATABASE MANAGEMENT SYSTEM
74COMPUTER CORPORATION OF AMERICA
CCA204 USER LANGUAGE REFERENCE MANUAL

  DATABASE MANAGEMENT SYSTEM
75BURROUGHS
DMSII DATA AND STRUCTURE DEFINITION LANGUAGE(DASDL)REFERE
     NCE MANUAL
  DBTG IMPLEMENTATION--DIFFERENCES REPRESENTATION OF INTE
       RFILE RELATIONSHIPS AND COMBINED LOGICAL AND PHYSICAL DEFINITIO
       N

  DATABASE MANAGEMENT SYSTEM
70CDC
MARS III REFERENCE MANUAL
  BOTH HOST AND SELF-CONTAINED FACILITIES

  DATABASE MANAGEMENT SYSTEM
70IBM
GIS VERSION 2 APPLICATION DESCRIPTION

  DATABASE MANAGEMENT SYSTEM
75UNIVAC
GLP USERS REFERENCE MANUAL

  DATABASE OPERATIONS
71COMPUVISOR
ASAP FILE MAINTENANCE AND INFORMATION RETRIEVAL SYSTEM RE
     FERENCE MANUAL
  INVERT PARTS OF ITEM AROUND COMMAS(REVERSE), TEST FOR C
      HANGE IN AN ITEMS VALUE DURING A RUN

  DATABASE OPERATORS
74HAMMER
#DATA ABSTRACTIONS FOR DATA BASES#
  HIGH LEVEL, APPLICATION ORIENTED

  DATABASE OPERATORS
74LINDGREEN
#BASIC OPERATIONS ON INFORMATION AS A BASIS FOR DATA BASE
      DESIGN#
  INCLUSION,EXTRACTION, CONGLOMERATION, SEPARATION, DERIV
       ATION, DETECTION, AND EXPOSITION--ALL WELL DEFINED IN TERMS OF
       FLAT FILES.

  DATABASE OPERATORS
75TSICHRITZIS
LSL: A LINK AND SELECTOR LANGUAGE

  DATABASE OPERATORS
74LEAVENWORTH + SAMMET
≠AN OVERVIEW OF NON PROCEDURAL LANGUAGES≠
  IN A HIGH LEVEL LANGUAGE

  DATABASE OPERATORS
74MINSKY
≠ON INTERACTION WITH DATA BASES≠
  PERMITTED USE

  DATABASE OPERATORS
74LISKOV + ZILLES
≠PROGRAMMING WITH ABSTRACT DATA TYPES≠
  ASSOCIATED WITH A DATA TYPE

  DATABASE OPERATORS
67U.S. DOD
≠OPERATIONAL CAPABILITY DESCRIPTION FOR A DATA MANAGEMENT
      SYSTEM≠
  FILE LEVEL AND RECORD LEVEL

  DATABASE SYSTEM DESIGN OBJECTIVES
74SNUGGS + POPEK + PETERSON
≠DATA BASE SYSTEM OBJECTIVES AS DESIGN CONSTRAINTS≠

  DATE SENSITIVE FILE
74RANDALL
≠INTEROGATING DATE-SENSITIVE FILES≠
  RECORDS CHANGES OVER TIME, NOT JUST SNAPSHOT OF CURRENT
       STATE

  DEADLOCK
73KING + COLLMEYER
≠DATABASE SHARING -- AN EFFICIENT MECHANISM FOR SUPPORTIN
    G CONCURRENT PROCESSES
  PREVENTION

  DESIGN DATABASE
76KAHN
≠A METHOD FOR DESCRIBING INFORMATION REQUIRED BY THE DATA
    BASE DESIGN PROCESS≠
  CONTENT

  DESIGN TRADEOFFS
74SNUGGS + POPEK + PETERSON
≠DATA BASE SYSTEM OBJECTIVES AS DESIGN CONSTRAINTS≠
  AMONG DATABASE SYSTEM OBJECTIVES, USER MUST GIVE SUFFIC
      IENT INFORMATION TO ENABLE TRADEOFFS TO BE MADE

  DEVICE INDEPENDENCE
72DANA + PRESSER
≠AN INFORMATION STRUCTURE FOR DATA BASE AND DEVICE INDEPE
    NDENT REPORT GENERATION≠
  IN REPORT DEFINITION
  ENCODE/DECODE
74COMPUTER CORPORATION OF AMERICA
CCA204 USER LANGUAGE REFERENCE MANUAL
  VARIABLE ITEMS ALWAYS ENCODED

  ENCODE/DECODE
73COMSHARE
COMPOSIT 77 USERS GUIDE

  ENCODE/DECODE
70BLEIER
≠DATA DEFINITION STANDARDIZATION≠
  AS PARK OF DATA ITEM DEFINITION

  ENCODE/DECODE
71COMPUVISOR
ASAP FILE MAINTENANCE AND INFORMATION RETRIEVAL SYSTEM RE
     FERENCE MANUAL
  EITHER DIRECTION AS SPECIFIED BY USER, RESIDUAL CODES,
       CODES ON RANGES OF VALUES

  ENCRYPTION
73COMSHARE
COMPOSIT 77 USERS GUIDE

  ENCRYPTION
71HOFFMAN
≠THE FORMULARY MODEL FOR FLEXIBLE PRIVACY AND ACCESS CONT
     ROLS≠
  VERBS TO USE IN A USER WRITTEN ACCESS CONTROL ROUTINE

  EVOLVABILITY
74SNUGGS + POPEK + PETERSON
≠DATA BASE SYSTEM OBJECTIVES AS DESIGN CONSTRAINTS≠

  EXCEPTION CONDITION ACTIONS
74MINSKY
≠ON INTERACTION WITH DATA BASES≠
  TAKEN WHEN CONSTRAINTS OR PREMISES FAIL

  EXCEPTION CONDITION RESPONSES
75GOODENOUGH
≠EXCEPTION HANDLING: ISSUES AND A PROPOSED NOTATION≠
  ABORT JOB AND ROLL-BACK, TRY AGAIN, NOTIFY INVOKER OF P
       ARTIAL RESULTS

  EXCEPTION CONDITION RESPONSES
71HOFFMAN
≠THE FORMULARY MODEL FOR FLEXIBLE PRIVACY AND ACCESS CONT
     ROLS≠
  TO ACCESS CONTROL VIOLATIONS. SYSTEM SHOULD NOT SAY TOO
        MUCH, IF ANYTHING, TO USER ATTEMPTING BREACH OF PRIVACY

  EXCEPTION CONDITIONS
74PROGRAM PRODUCTS
THE DATA ANALYZER TECHNICAL REFERENCE MANUAL
  MISMATCH OPTION WITH COORDINATED FILES FOR MISSING MAST
      ER OR SLAVE RECORD

  EXCEPTION CONDITIONS
75GOODENOUGH
≠EXCEPTION HANDLING: ISSUES AND A PROPOSED NOTATION≠
```

```
  RANGE FAILURE, DOMAIN FAILURE, RESULTS, OR MONITORING

  EXCEPTION PROCESSING
75GOODENOUGH
#EXCEPTION HANDLING: ISSUES AND A PROPOSED NOTATION#
  SYSTEM DEFAULTS OR USER SPECIFIED

  FEATURE LIST
75WEBER
#AUDIT CAPABILITIES OF SOME DATABASE MANAGEMENT SYSTEMS#
  FOR DMS

  FEATURE LIST
71CODASYL
FEATURE ANALYSIS OF GENERALIZED DATA BASE MANAGEMENT SYST
     EMS

  FILE LEVEL DERIVATION
73COMSHARE
COMPOSIT 77 USERS GUIDE
  SUCH AS STATISTICAL OUTPUT, CORRELATION, AND REGRESSION

  FLAT FILE
75COMPUTER ASSOCIATES
EARL GENERAL INFORMATION MANUAL

  FLOW DATA
74TAGGART
#DEVELOPING AN ORGANIZATION#S INFORMATION INVENTORY#
  CHANGE OVER TIME PERIOD, RATE OF CHANGE, EVENT INFORMAT
       ION

  FORMATION AND PRESENTATION
73COMSHARE
COMPOSIT 77 USERS GUIDE
  TABULAR, TWO DIMENSIONAL, GRAPHICAL

  FUNCTIONS
67U.S. DOD
#OPERATIONAL CAPABILITY DESCRIPTION FOR A DATA MANAGEMENT
      SYSTEM#
  ON DATABASE: DEFINITION, MAINTENANCE, RETRIEVAL, RESTRU
       CTUREING, OUTPUT

  GRAPHICAL OUTPUT
73COMSHARE
COMPOSIT 77 USERS GUIDE

  HIERARCHICAL SELECTION CRITERIA
67U.S. DOD
#OPERATIONAL CAPABILITY DESCRIPTION FOR A DATA MANAGEMENT
      SYSTEM#

  HIGH LEVEL LANGUAGE
74LEAVENWORTH + SAMMET
#AN OVERVIEW OF NON PROCEDURAL LANGUAGES#
  PROPERTIES OF, INCLUDING ELIMINATION OF ARBITRARY SEQUE
       NCES AND USE OF AGGREGATE OPERATORS ON WHOLE DATA STRUCTURE

  HIGH LEVEL MULTIFILE LANGUAGE
74OLLE
#SELF CONTAINED FACILITIES TO PROCESS SUB STRUCTURES OF M
```

```
    ORE COMPLEX NETWORK STRUCTURES≠
  DESIGN CONSIDERATIONS FOR NON-PROGRAMMING USER INTERFAC
      E

  HIGH LEVEL MULTIFILE LANGUAGE
75UNIVAC
GLP USERS REFERENCE MANUAL

  HIGH LEVEL MULTIFILE LANGUAGE
75TSICHRITZIS
LSL: A LINK AND SELECTOR LANGUAGE

  HIGH LEVEL MULTIFILE LANGUAGE
76ASTRAHAN
≠SYSTEM R: RELATIONAL APPROACH TO DATABASE MANAGEMENT≠
  SEQUEL FOR QUERY AND UPDATE

  INDEX DEFINITION
70CDC
MARS III REFERENCE MANUAL

  INFERENTIAL RETRIEVAL
74SKINNER
≠A HEURISTIC APPROACH TO INDUCTIVE INFERENCE IN FACT RETR
    IEVAL SYSTEMS≠
  FOR DERIVING MORE THAN STORED FACTS

  INFORMATION ALGEBRA
72TEICHROEW
≠A SURVEY OF LANGUAGES FOR STATING REQUIREMENTS FOR COMPU
    TER-BASED INFORMATION SYSTEMS≠

  INFORMATION ALGEBRA
74LINDGREEN
≠BASIC OPERATIONS ON INFORMATION AS A BASIS FOR DATA BASE
    DESIGN≠

  INFORMATION ALGEBRA
74TAGGART
≠DEVELOPING AN ORGANIZATION≠S INFORMATION INVENTORY≠

  INFORMATION REQUIREMENTS
76KAHN
≠A METHOD FOR DESCRIBING INFORMATION REQUIRED BY THE DATA
    BASE DESIGN PROCESS≠

  INFORMATION SYSTEM MODELLING
72SENKO
≠DETAILS OF A SCIENTIFIC APPROACH TO INFORMATION SYSTEMS≠
  PRIMARILY AT THE PHYSICAL LEVEL OF A DATA STRUCTURE, AC
      CESS ROUTINES, HARDWARE, AND REQUESTS

  INFORMATION SYSTEM REQUIREMENTS SPECIFICATIONS
72TEICHROEW
≠A SURVEY OF LANGUAGES FOR STATING REQUIREMENTS FOR COMPU
    TER-BASED INFORMATION SYSTEMS≠

  INFORMATION SYSTEM REQUIREMENTS SPECIFICATIONS
72SENKO
≠DETAILS OF A SCIENTIFIC APPROACH TO INFORMATION SYSTEMS≠
  IN TERMS OF APPLICATIONS, DATA STRUCTURE, AND HARDWARE
  INFORMATION SYSTEM REQUIREMENTS STATEMENT LANGUAGE
72TEICHROEW
≠A SURVEY OF LANGUAGES FOR STATING REQUIREMENTS FOR COMPU
     TER-BASED INFORMATION SYSTEMS≠
  COMPARISON OF SEVERAL INCLUDING ADS, TAG, INFORMATION A
       LGEBRA, LANGEFORS, LOMBARDI, AND YOUNG AND KENT

  INTERROGATION
71CODASYL
FEATURE ANALYSIS OF GENERALIZED DATA BASE MANAGEMENT SYST
     EMS

  LOCKS ON DATA
73KING + COLLMEYER
≠DATABASE SHARING -- AN EFFICIENT MECHANISM FOR SUPPORTIN
     G CONCURRENT PROCESSES
  FOR CONCURRENT PROCESSES

  LOGICAL PAGE SIZE
72DANA + PRESSER
≠AN INFORMATION STRUCTURE FOR DATA BASE AND DEVICE INDEPE
     NDENT REPORT GENERATION≠
  MAPPED TO PHYSICAL DEVICE PAGE SIZE

  OBJECTIVES
74SNUGGS + POPEK + PETERSON
≠DATA BASE SYSTEM OBJECTIVES AS DESIGN CONSTRAINTS≠
  IN DESIGN OF DATABASE SYSTEM

  PATTERNS OF PROCESSING
72TEICHROEW
≠A SURVEY OF LANGUAGES FOR STATING REQUIREMENTS FOR COMPU
     TER-BASED INFORMATION SYSTEMS≠
  HIGH LEVEL LANGUAGE FOR EXPRESSING

  PATTERNS OF PROCESSING
74LEAVENWORTH + SAMMET
≠AN OVERVIEW OF NON PROCEDURAL LANGUAGES≠
  FUNCTIONS IN INFORMATION SYSTEM PERFORMED BASED UPON A
       PATTERN IN THE DATA

  PATTERNS OF PROCESSING
74MINSKY
≠ON INTERACTION WITH DATA BASES≠
  PRIMITIVE OPERATIONS

  PERFORMANCE CRITERIA
74SNUGGS + POPEK + PETERSON
≠DATA BASE SYSTEM OBJECTIVES AS DESIGN CONSTRAINTS≠
  TIME, SPACE, VOLATILITY, RESPONSE TIMES

  PERFORMANCE MONITORING
74SNUGGS + POPEK + PETERSON
≠DATA BASE SYSTEM OBJECTIVES AS DESIGN CONSTRAINTS≠

  PERFORMANCE MONITORING
67U.S. DOD
≠OPERATIONAL CAPABILITY DESCRIPTION FOR A DATA MANAGEMENT
     SYSTEM≠
  DISCUSSES SEVERAL OPTIONS

  PREMISE ACTION
```

```
76DIJKSTRA
A DISCIPLINE OF PROGRAMMING

  PREMISE ACTION
71CODASYL
FEATURE ANALYSIS OF GENERALIZED DATA BASE MANAGEMENT SYST
     EMS

  PROCESS DEFINITION
76KAHN
≠A METHOD FOR DESCRIBING INFORMATION REQUIRED BY THE DATA
     BASE DESIGN PROCESS≠
  RECORD-LEVEL VERBS, BEHAVIORAL CHARACTERISTICS

  PROCESS DEFINITION
74PROGRAM PRODUCTS
THE DATA ANALYZER TECHNICAL REFERENCE MANUAL
  SEQUENTIAL, PROCEDURAL, FORMS DRIVEN

  PROCESS DEFINITION
69LYNCH
≠ADS:A TECHNIQUE IN SYSTEM DOCUMENTATION≠
  PREMISE: ACTION IN DECISION TABLE FORM

  PROGRAM GENERATION
73NUNAMAKER + SWENSON + WHINSTON
≠SPECIFICATIONS FOR THE DEVELOPMENT OF A GENERALIZED DATA
      BASE PLANNING SYSTEM≠
  IN RESPONSE TO INQUIRY OR REQUEST

  PSL
72TEICHROEW
≠A SURVEY OF LANGUAGES FOR STATING REQUIREMENTS FOR COMPU
     TER-BASED INFORMATION SYSTEMS≠

  QUERY CONDITIONS
71CODASYL
FEATURE ANALYSIS OF GENERALIZED DATA BASE MANAGEMENT SYST
     EMS

  QUERY RESPONSES
74SKINNER
≠A HEURISTIC APPROACH TO INDUCTIVE INFERENCE IN FACT RETR
     IEVAL SYSTEMS≠

  QUERY TYPES
74PROGRAM PRODUCTS
THE DATA ANALYZER TECHNICAL REFERENCE MANUAL

  QUERY TYPES
74SKINNER
≠A HEURISTIC APPROACH TO INDUCTIVE INFERENCE IN FACT RETR
     IEVAL SYSTEMS≠

  RECOVERY
73KING + COLLMEYER
≠DATABASE SHARING -- AN EFFICIENT MECHANISM FOR SUPPORTIN
     G CONCURRENT PROCESSES

  RECOVERY PROCEDURES
74SAYANI
≠RESTART AND RECOVERY IN A TRANSACTION-ORIENTED INFORMATI
```

```
    ON PROCESSING SYSTEM≠
  FOR EACH TYPE OF DATA LOSS

  REPORT DEFINITION
74TAGGART
≠DEVELOPING AN ORGANIZATION≠S INFORMATION INVENTORY≠
  USING CONCEPT OF INFORMATION ELEMENT

  REPORT DEFINITION
69LYNCH
≠ADS:A TECHNIQUE IN SYSTEM DOCUMENTATION≠
  FORM OF OUTPUT AND SOURCE OF DATA CONTENT

  REPORT DEFINITION
74COMPUTER CORPORATION OF AMERICA
CCA204 USER LANGUAGE REFERENCE MANUAL

  REPORT DEFINITION
72DANA + PRESSER
≠AN INFORMATION STRUCTURE FOR DATA BASE AND DEVICE INDEPE
     NDENT REPORT GENERATION≠
  FOCUSES ON LOGICAL USER SPECIFICATIONS

  REPORT DEFINITION
67U.S. DOD
≠OPERATIONAL CAPABILITY DESCRIPTION FOR A DATA MANAGEMENT
     SYSTEM≠

  REPORT DEFINITION
74PROGRAM PRODUCTS
THE DATA ANALYZER TECHNICAL REFERENCE MANUAL
  DEFAULTS OR USER OVERRIDE

  REPORT DEFINITION
75COMPUTER ASSOCIATES
EARL GENERAL INFORMATION MANUAL
  DEFAULT LABELING AND EDITING. VALUE DEPENDENT OUTPUT CO
       NTROL

  REPORT GENERATION
67U.S. DOD
≠OPERATIONAL CAPABILITY DESCRIPTION FOR A DATA MANAGEMENT
     SYSTEM≠
  ON DEMAND OR TRIGGERED BY SPECIFIED CONDITION

  REPORT GENERATION
70CDC
MARS III REFERENCE MANUAL
  WHEN REPORTS GENERATED OR TRIGGERED

  RESTART
74SAYANI
≠RESTART AND RECOVERY IN A TRANSACTION-ORIENTED INFORMATI
    ON PROCESSING SYSTEM≠

  RETENTION
69LYNCH
≠ADS:A TECHNIQUE IN SYSTEM DOCUMENTATION≠
  OF MASTER FILE DATA

  RULES OF INFERENCE
74SKINNER
```

```
≠A HEURISTIC APPROACH TO INDUCTIVE INFERENCE IN FACT RETR
     IEVAL SYSTEMS≠
  SIMILARITY STRUCTURE, LEVELS OF SIGNIFICANCE

  SECURITY
73FOSTER
≠DATA SECURITY DEVELOPMENTS AT IBM≠
  ENVIRONMENTS--(1)NATIONAL SECURITY, (2)INTERNAL CORPORA
        TE MIS, AND (3)COMMERCIAL PRIVACY(COMPUTER UTILITY). USER SPECI
        FIES LEVEL OF SECURITY OR CONSEQUENCES IF DATA COMPROMISED.

  SECURITY MONITORING
71CONWAY + MAXWELL + MORGAN
≠ON THE IMPLEMENTATION OF SECURITY MEASURES IN INFORMATIO
    N SYSTEMS≠
  KEEP A LOG OF UNSUCCESSFUL ACCESS ATTEMPTS

  SELECTION CRITERIA
74RANDALL
≠INTEROGATING DATE-SENSITIVE FILES≠
  BASED ON TIME POINT(AT, BEFORE, AFTER, EVER) OR PERIOD(
        IN, DURING, UNTIL, SINCE, ALWAYS)

  SELECTION CRITERIA
74COMPUTER CORPORATION OF AMERICA
CCA204 USER LANGUAGE REFERENCE MANUAL
  ON ITEM VALUES OR RANGES. MULTIVALUED ITEM DEFAULT ONLY
         FIRST INSTANCE, OTHERWISE ANY INSTANCE

  SELECTION CRITERIA
67U.S. DOD
≠OPERATIONAL CAPABILITY DESCRIPTION FOR A DATA MANAGEMENT
      SYSTEM≠
  LANGUAGE ELEMENTS ON HIERARCHICAL STRUCTURE

  SELECTION CRITERIA
74OLLE
≠SELF CONTAINED FACILITIES TO PROCESS SUB STRUCTURES OF M
    ORE COMPLEX NETWORK STRUCTURES≠
  SHOULD BE SAME IN UPDATE AND QUERY

  SEMANTIC CONSTRAINTS
76ASTRAHAN
≠SYSTEM R: RELATIONAL APPROACH TO DATABASE MANAGEMENT≠
  CALLED ASSERTIONS

  SEMANTIC CONSTRAINTS
75ESWARAN + CHAMBERLIN
≠FUNCTIONAL SPECIFICATIONS OF A SUBSYSTEM FOR DATA BASE I
    NTEGRITY≠
  AS VALIDATION CRITERIA

  SEMANTIC INTEGRITY
75HAMMER + MCLEOD
≠SEMANTIC INTEGRITY IN A RELATIONAL DATA BASE SYSTEM≠
  DISCUSSES TWO TYPES-STATE AND TRANSITION

  STATUS DATA
74TAGGART
≠DEVELOPING AN ORGANIZATION≠S INFORMATION INVENTORY≠
  LEVEL OR STATE AT A POINT IN TIME
  STRING ITEM TYPE
70IBM
GIS VERSION 2 APPLICATION DESCRIPTION
  AND SUBSTRING SEARCH

  SURVEY OF DMS
71CODASYL
FEATURE ANALYSIS OF GENERALIZED DATA BASE MANAGEMENT SYST
     EMS

  SURVEY OF DMS
75WEBER
≠AUDIT CAPABILITIES OF SOME DATABASE MANAGEMENT SYSTEMS≠
  COMPARES 20 SYSTEMS

  SYSTEM DEVELOPMENT PHASES
72TEICHROEW
≠A SURVEY OF LANGUAGES FOR STATING REQUIREMENTS FOR COMPU
     TER-BASED INFORMATION SYSTEMS≠
  ANALYSIS, DESIGN, AND CONSTRUCTION

  TEXT ITEM TYPE
71COMPUVISOR
ASAP FILE MAINTENANCE AND INFORMATION RETRIEVAL SYSTEM RE
     FERENCE MANUAL
  WITH CONCATENATION, AND COMPRESSION OPERATORS

  TIME
74TAGGART
≠DEVELOPING AN ORGANIZATION≠S INFORMATION INVENTORY≠
  ANOTHER DIMENSION ON THE STORED DATABASE

  TIME STAMPING
76GRIFFITHS + WADE
≠AN AUTHORIZATION MECHANISM FOR A RELATIONAL DATABASE SYS
     TEM≠
  FOR GRANTING AND REVOKING PRIVILEGES

  TIME STAMPING
74RANDALL
≠INTERROGATING DATE-SENSITIVE FILES≠
  ON STORED DATA ITEMS IN RECORDS AND REPEATING GROUPS AN
       D USED IN SELECTION CRITERIA. EFFECTIVE DATE-WHEN CHANGE OCCURS
        IN REAL WORLD, NOT WHEN DATA ENTERED

  TRADEOFFS
74OLLE
≠SELF CONTAINED FACILITIES TO PROCESS SUB STRUCTURES OF M
     ORE COMPLEX NETWORK STRUCTURES≠
  MADE BY USER OR SYSTEM. USER KNOWLEDGE FOR TRADEOFF CAN
        BE MANDATORY, OPTIONAL, OR HIDDEN.

  TRANSACTION DEFINITION
76ASTRAHAN
≠SYSTEM R: RELATIONAL APPROACH TO DATABASE MANAGEMENT≠
  A SET OF SEQUEL COMMANDS

  TRANSACTION DEFINITION
69LYNCH
≠ADS:A TECHNIQUE IN SYSTEM DOCUMENTATION≠
  NAME, TYPE, SIZE, VALIDATION CRITERIA
  TRANSACTION DEFINITION
72SENKO
≠DETAILS OF A SCIENTIFIC APPROACH TO INFORMATION SYSTEMS≠

  TRANSACTION PROCESSING
72SENKO
≠DETAILS OF A SCIENTIFIC APPROACH TO INFORMATION SYSTEMS≠
  SPECIFICATION OF FOR SYSTEM DESIGN

  TRANSACTION PROCESSING
71CODASYL
FEATURE ANALYSIS OF GENERALIZED DATA BASE MANAGEMENT SYST
    EMS

  TRANSACTION VALIDATION CRITERIA
69LYNCH
≠ADS:A TECHNIQUE IN SYSTEM DOCUMENTATION≠

  TRIGGER
76ASTRAHAN
≠SYSTEM R: RELATIONAL APPROACH TO DATABASE MANAGEMENT≠

  UPDATE
71COMPUVISOR
ASAP FILE MAINTENANCE AND INFORMATION RETRIEVAL SYSTEM RE
    FERENCE MANUAL
  INCLUDES ACTIVATE AND DEACTIVATE OPERATIONS

  UPDATE
71CODASYL
FEATURE ANALYSIS OF GENERALIZED DATA BASE MANAGEMENT SYST
    EMS

  UPDATE PROCESSES
70IBM
GIS VERSION 2 APPLICATION DESCRIPTION
  DEPENDENT UPON DATA EXISTENCE

  UPDATE PROCESSES
73KING + COLLMEYER
≠DATABASE SHARING -- AN EFFICIENT MECHANISM FOR SUPPORTIN
    G CONCURRENT PROCESSES
  CONCURRENCY CONTROL

  UPDATE PROCESSES
74COMPUTER CORPORATION OF AMERICA
CCA204 USER LANGUAGE REFERENCE MANUAL
  OPERATORS TO CHANGE, ADD OR DELETE FIELDS OR RECORDS

  USERSCHEMA
75TSICHRITZIS
LSL: A LINK AND SELECTOR LANGUAGE
  USER VIEW OF DATA

  USERSCHEMA
76ASTRAHAN
≠SYSTEM R: RELATIONAL APPROACH TO DATABASE MANAGEMENT≠

  USERSCHEMA
76GRIFFITHS + WADE
≠AN AUTHORIZATION MECHANISM FOR A RELATIONAL DATABASE SYS
    TEM≠
  DEFINED AS VIEWS TO A RELATIONAL DATABASE. BASIS FOR AC
        CESS CONTROL

  VALIDATION
75TSICHRITZIS
LSL: A LINK AND SELECTOR LANGUAGE

  VALIDATION CRITERIA
76ASTRAHAN
*SYSTEM R: RELATIONAL APPROACH TO DATABASE MANAGEMENT*
  ACCEPTABLE STATES OR TRANSITIONS. ENFORCED AT END OF TR
        ANSACTION OR WHEN USERINVOKES THEM. ROLL BACK WHEN FAILURE

  VALIDATION CRITERIA
74SKINNER
*A HEURISTIC APPROACH TO INDUCTIVE INFERENCE IN FACT RETR
      IEVAL SYSTEMS*
  CAN BE USED TO DETECT INCONSISTENCIES WHERE DEFINED SIM
        ILARITY STRUCTURES INDICATE OTHERWISE

  VALIDATION CRITERIA
67U.S. DOD
*OPERATIONAL CAPABILITY DESCRIPTION FOR A DATA MANAGEMENT
      SYSTEM*
  RANGE, ENUMERATION, TABLE LOOK UP, SEQUENCE, FORMAT, DA
        TA DEPENDENT

  VALIDATION CRITERIA
75ESWARAN + CHAMBERLIN
*FUNCTIONAL SPECIFICATIONS OF A SUBSYSTEM FOR DATA BASE I
      NTEGRITY*
  TO WHAT THEY APPLY, HOW THEY APPLY, WHEN THEY APPLY, AN
        D WHEN TO INVOKE THEM

  VALIDATION CRITERIA
74MINSKY
*ON INTERACT. N WITH DATA BASES*
  DATABASE

  VALIDATION CRITERIA
75HAMMER + MCLEOD
*SEMANTIC INTEGRITY IN A RELATIONAL DATA BASE SYSTEM*
  STATE-VALID DOMAINS AND RELATIONS CHECKED. TRANSITION-L
        EGAL OPERATORS. LANGUAGE TO STATE AND FUNCTION TO STORE CRITERI
        A

  VALIDATION PROCESS
75HAMMER + MCLEOD
*SEMANTIC INTEGRITY IN A RELATIONAL DATA BASE SYSTEM*
  TO TEST STORED CRITERIA AGAINST DATA AND INITIATE EXCEP
        TION RESPONSE

  VALIDATION RESPONSE
75ESWARAN + CHAMBERLIN
*FUNCTIONAL SPECIFICATIONS OF A SUBSYSTEM FOR DATA BASE I
      NTEGRITY*
  WHEN VALIDATION CRITERIA VIOLATED--HARD(REJECT TRANSACT
        ION)OR SOFT (ISSUE MESSAGE TO USER)

  VALIDATION RESPONSE
75HAMMER + MCLEOD
*SEMANTIC INTEGRITY IN A RELATIONAL DATA BASE SYSTEM*
```

4. BIBLIOGRAPHIC ENTRIES WITH ANNOTATIONS

```
ID YYAUTHOR-* 67U.S. DOD
TITLE-------* #OPERATIONAL CAPABILITY DESCRIPTION FOR A DATA MANAGEMENT
      SYSTEM#
MONOGRAPH TITLE----* WORLD WIDE MILITARY COMMAND CONTROL SYSTEM (WWMCCS
     )-APPENDIX TO RFP
YEAR---------------* 1967

  AUTHOR NAME-* U.S. DEPARTMENT OF DEFENSE

  DESCRIPTOR--* DATA STRUCTURE MODEL
  DESCRIPTION-* (1)SINGLE PATH HIERARCHY,(2)MULTI-PATH HIERARCHY,(3)INV
        ERTED HIERARCHY (MULTIPLE PARENTS),(4)LINKED HIERARCHY(NETWORK)
        ,(5)LINKED LIST(POINTERS ON INSTANCES), AND(6)ASSOCIATIVE LINK(
        BY CONTENT)

  DESCRIPTOR--* VALIDATION CRITERIA
  DESCRIPTION-* RANGE, ENUMERATION, TABLE LOCK UP, SEQUENCE, FORMAT, DA
        TA DEPENDENT

  DESCRIPTOR--* SELECTION CRITERIA
  DESCRIPTION-* LANGUAGE ELEMENTS ON HIERARCHICAL STRUCTURE

  DESCRIPTOR--* HIERARCHICAL SELECTION CRITERIA

  DESCRIPTOR--* REPORT DEFINITION

  DESCRIPTOR--* REPORT GENERATION
  DESCRIPTION-* ON DEMAND OR TRIGGERED BY SPECIFIED CONDITION

  DESCRIPTOR--* BEHAVIORAL CHARACTERISTICS
  DESCRIPTION-* OF FILES, DYNAMICS, GROWTH, CHANGE

  DESCRIPTOR--* FUNCTIONS
  DESCRIPTION-* ON DATABASE: DEFINITION, MAINTENANCE, RETRIEVAL, RESTRU
        CTUREING, OUTPUT

  DESCRIPTOR--* DATABASE OPERATORS
  DESCRIPTION-* FILE LEVEL AND RECORD LEVEL

  DESCRIPTOR--* PERFORMANCE MONITORING
  DESCRIPTION-* DISCUSSES SEVERAL OPTIONS

  DESCRIPTOR--* DATA DEFINITION
  DESCRIPTION-* INCLUDING OPERATIONS
```

ID YYAUTHOR: 67U.S. DOD

TITLE: "Operational Capability Description for a Data Management System" (WWMCCS)

This document is the appendix to the RFP for WWMCCS (World Wide Military Command Control System).

The system should be able to handle six data structures (it is not clear whether the relationships in 1, 2, and 3 apply to group types or group instances):

1. simple hierarchy -- single path hierarchy (1 parent, 1 dependent)
2. hierarchy -- tree structure or multi-path hierarchy (1 parent, many dependents)
3. inverted hierarchy -- 1 subordinate, many parents
4. hierarchical linked -- network
5. linked list -- physically linked group repetitions with pointers
6. associative linking -- links based on content

The data element is the basic data item, with no substructure. Data elements are collected into groups which may or may not be repeating.

The WWMCCS calls for an extensive range of validity checks including: a range of values, an explicit set of values, table look up, sequence checks, format consistency, and cross comparison (i.e., comparison of an element value with other values in the data base for consistency).

For selection, qualification should include:

1. group where an item meets a specified boolean selection expression (BSE),
2. superior where EVERY subordinate meets BSE,
3. superior where ANY subordinate meets BSE,
4. all subordinates of a qualified group,
5. superior based on qualification of subordinate n levels removed,
6. any groups relating to group (element) meeting BSE,
7. groups meeting secondary BSE when primary BSE not met by any group.

ID YYAUTHOR: 67U.S. DOD

TITLE: "Operational Capability Description for a Data Management System" (WWMCCS)

For presentation of results report generation and definition capabilities including; automatic row and column spacing, paging, titles, headers, footings, and data editing. Reports should be named and produced on demand or when triggered by specified conditions.

WWMCCS calls for an overlapping seven way classification of files. The overlap is because some of the dimensions related to behavior, some to processes on the files, and some to data definition. The seven classifications are:

1. very large files
2. dynamically changing (maintenance oriented)
3. static files
4. growth files
5. unformatted text files
6. small data element files (bits/bytes)
7. retrieval oriented

Categories five and six are definitional since they relate to data format. Category one is one side of a size dimension which relates to behavior. The other categories (2, 3, 4, 7) are fuzzy since they overlap on several dimensions and relate to both behavior and process. Maintenance and retrieval are opposite extremes. Static may refer to either content or size, i.e., no changes or only value modification changes.

Operations on the database are classified in three ways. First, five types of file processing are defined:

1. definiton
2. maintenance (both value modification and insert/delete)
3. retrieval
4. restructuring
5. output

Second, they define four types of functions:

1. insert }
2. delete } no distinction is made for element, group, or entry

ID YYAUTHOR: 67U.S. DOD

TITLE: "Operational Capability Description for a Data Management System" (WWMCCS)

3. change -- value modification on a uniquely identified record instance
4. mass update -- stated with respect to changes, but could also be applicable for inserts and deletes

The third set of classifications can be inferred from their five types of benchmark jobs:

1. single file update
2. update and report generation
3. multifile report generation
4. multifile update
5. online retrieval

The system requires extensive performance monitoring capability, with the option to turn the monitoring on or off on demand. Also certain system status information should be available to the user, who should have the option of suspending his job and continuing it later.

```
ID YYAUTHOR-* 69LYNCH
TITLE-------* *ADS:A TECHNIQUE IN SYSTEM DOCUMENTATION*
SERIAL TITLE-------* DATA BASE
SERIAL NUMBER------* 1:1
PUBLISHER----------* ACM
YEAR---------------* 1969
DATE---------------* 1969 SPRING
PAGES--------------* 6-18

   AUTHOR NAME-* LYNCH
   FIRST NAME--* HUGH J.

   DESCRIPTOR--* REPORT DEFINITION
   DESCRIPTION-* FORM OF OUTPUT AND SOURCE OF DATA CONTENT

   DESCRIPTOR--* TRANSACTION DEFINITION
   DESCRIPTION-* NAME, TYPE, SIZE, VALIDATION CRITERIA

   DESCRIPTOR--* TRANSACTION VALIDATION CRITERIA

   DESCRIPTOR--* DATA ITEM DERIVATION RULES

   DESCRIPTOR--* RETENTION
   DESCRIPTION-* OF MASTER FILE DATA

   DESCRIPTOR--* PROCESS DEFINITION
   DESCRIPTION-* PREMISE: ACTION IN DECISION TABLE FORM

   DESCRIPTOR--* DATA DEFINITION
   DESCRIPTION-* FLAT FILES AND COORDINATED FILES
```

ID YYAUTHOR: 69LYNCH

TITLE: "ADS: A Technique in Systems Documentation"

ADS is a concise, cross-referenced system for definition of system specifications and communications among users, systems analysts, and programmers. A forms (report, input, computational, history, and logic) driven system for capturing much data and process defining information and some limited behavioral data.

1. Report -- defines reports by specifying headers, data items, editing rules, and vertical and horizontal positioning information. Cross-reference to where items defined (input, computational, or history).

2. Input -- defines input records with item name, position, size, A/N (alpha/numeric type and implied left and right justification), validation criteria (premises only), percent of the records for which the item is present, and a flag to indicate if a transaction code needed to identify items. Assumes set of independent items in a flat file.

3. Computational -- provides rules for generating derived items. Specifies derived item name, two operands (with a cross-reference to their original definition) and an operator.

4. Historical -- defines Master File records. Item name, percent of records where item occurs, A/N, maximum size, retention (how long to retain), and cross-reference to original definition.

5. Logic -- premise - action definition in decision table form

Data definition (combination of logical and physical) provided on input, historical, and computation forms. Assumes single type of input and single historical record formats in flat file structure. Report form provides report definition.

ID YYAUTHOR: 69LYNCH

TITLE: "ADS: A Technique in Systems Documentation"

Processes are defined in the computational and logic forms and implied in the report form.

Behavioral data only as percent of records in which item occurs. When occurrence 100 percent, no system implication that the item is mandatory.

```
ID YYAUTHOR-* 70BLEIER
TITLE--------* ≠DATA DEFINITION STANDARDIZATION≠
SERIAL TITLE--------* PROCEEDINGS SIGFIDET WORKSHOP ON DATA DESCRIPTION
     AND ACCESS
PUBLISHER------------* ACM
YEAR-----------------* 1970
DATE-----------------* 1970/11
PAGES----------------* 60-66

  AUTHOR NAME-* BLEIER
  FIRST NAME--* ROBERT E.

  DESCRIPTOR--* DATA ITEM DEFINITION
  DESCRIPTION-* LIST 20 ATTRIBUTES, MOSTLY PHYSICAL, INCLUDING NAME, SI
        ZE, CODING TYPE, KEY, EDITING, ENCODING, DECODING, ACCESS CONTR
        OLS, AND VALIDATION CRITERIA

  DESCRIPTOR--* ENCODE/DECODE
  DESCRIPTION-* AS PARK OF DATA ITEM DEFINITION
```

ID YYAUTHOR: 70BLEIER

TITLE: "Data Definition Standardization"

Early work calling for data definition standards. All business data processing applications (whether batch or online and management or operations oriented) should share the same data and therefore need a common data definition. Common definition should also apply for the many DMS that are being developed. Once his data is defined the user should only need to specify what he wants done, not how to do it. Lists a set of twenty candidates for inclusion in data definition standard (most of them relate to physical rather than logical structure): (1) file name; (2) record length; (3) blocking; (4) record structure variability (fixed or variable length); (5) auxiliary storage device; (6) access methods; (7) tape label option (standard or special); (8) record type of segment names (hierarchic structure); (9) field name; (10) starting character (position); (11) field size; (12) coding type (real, binary, etc.); (13) key/non-key field; (14) editing rules; (15) networks (defining relationships independent of how they are implemented); (16) encoding (more for storage efficiency than encryption for protection); (17) decoding; (18) security access control; (19) data validation criteria; and (20) dictionary catalog (at file level).

```
ID YYAUTHOR-* 70CDC
TITLE-------* MARS III REFERENCE MANUAL
PUBLISHER----------* CDC
PUBLISHER LOCATION-* MINNEAPOLIS, MINNESOTA
YEAR---------------* 1970
DATE---------------* 1970/09

   AUTHOR NAME-* CONTROL DATA
   FIRST NAME--* CORP.

   DESCRIPTOR--* DATABASE MANAGEMENT SYSTEM
   DESCRIPTION-* BOTH HOST AND SELF-CONTAINED FACILITIES

   DESCRIPTOR--* DATA DEFINITION

   DESCRIPTOR--* INDEX DEFINITION

   DESCRIPTOR--* REPORT GENERATION
   DESCRIPTION-* WHEN REPORTS GENERATED OR TRIGGERED

   DESCRIPTOR--* ACCESS CONTROL
   DESCRIPTION-* PRIVACY LEVELS
```

ID YYAUTHOR: 70CDC
TITLE: MARS III Reference Manual

A Mars III datafile is a single hierarchial file structure. The system can handle repeating groups.

Data definition includes file, data item index and encode/decode table definitions. File definition includes privacy codes, unique valued primary index item, character for missing data, a record compression option, percent of records containing non-blank data, estimated number of records, space allocation for growth.

Data item definition includes maximum number of occurrences of a repeating group and a privacy code with a level of privacy.

Index definition includes the form of the index (indexing method), is index multiple item, when index is being defined (before or after database creation), estimated number of null values.

A subschema can be defined by permuting and selecting data items.

The report writer allows reports to be printed at a certain time (daily, weekly, at an interval, or on a specific date). To support this a user can build a calendar specifying year, leap years, Sat., Sun., workdays, and holidays.

Privacy is at many levels. Users have an identifier and a password. Data items have a privacy level number. Each user also has a privacy level number. A user can only access those items with equal or lower level numbers. A need to know flag is available to override privacy levels. It is associated with a user and data items. In this way a user can access data items of a higher privacy level.

```
ID YYAUTHOR-* 70IBM
TITLE-------* GIS VERSION 2 APPLICATION DESCRIPTION
PUBLISHER----------* IBM CORP

  AUTHOR NAME-* IBM

  DESCRIPTOR--* DATABASE MANAGEMENT SYSTEM

  DESCRIPTOR--* DATA DEFINITION
  DESCRIPTION-* INCLUDES DEFAULTS ON OUTPUT---COLUMN HEADING, EDITING,
       AND ENCODE-DECODE TABLE

  DESCRIPTOR--* STRING ITEM TYPE
  DESCRIPTION-* AND SUBSTRING SEARCH

  DESCRIPTOR--* ACCESS CONTROL
  DESCRIPTION-* FILE AND ITEM LEVEL. ITEM SECURITY CODES ATTACHED TO EA
       CH USER PASSWORD

  DESCRIPTOR--* UPDATE PROCESSES
  DESCRIPTION-* DEPENDENT UPON DATA EXISTENCE
```

ID YYAUTHOR: 70IBM

TITLE: GIS/2 Application Description

GIS is a DMS which supports hierarchically structured files and offers a procedural query language.

Data definition permits column headings, editing formats, synonyms for files and data items, and an encode/decode table.

Selection clause offers substring search for a string in a fixed location of an item or anywhere in the item, inter-record operators which determine if an item has increased, decreased or changed in value from the previous record and a test for empty or absent data in an item.

Security is at the file and item type level. Each item is assigned a security code which is also assigned to each password. A user may only use those items having the security codes assigned to his password. A file is assigned a password and can only be used if the user knows the password.

File updating is a function of the data existing or not existing. If a duplicate record is being added the user can ignore it or add the duplicate. When replacing a record the user can ignore update if record doesn't exist, replace record only if it exists or add or replace whether it doesn't or does exist. Item updates follow the same rules.

GIS provides for transaction processing. It handles file reading and searching, and field mapping. A transaction is read and the master file is searched for a matched record on that key. The user, in a GIS program, controls how the update is processed.

```
ID YYAUTHOR-* 71CODASYL
TITLE-------* FEATURE ANALYSIS OF GENERALIZED DATA BASE MANAGEMENT SYST
     EMS
PUBLISHER----------* ACM
YEAR---------------* 1971
DATE---------------* 1971 MAY
PAGES--------------* 520, INDEXED

   AUTHOR NAME-* CODASYL SYSTEMS COMMITTEE
   FIRST NAME--* T. WILLIAM OLLE, CHAIRMAN.

   DESCRIPTOR--* SURVEY OF DMS

   DESCRIPTOR--* FEATURE LIST

   DESCRIPTOR--* INTERROGATION

   DESCRIPTOR--* UPDATE

   DESCRIPTOR--* TRANSACTION PROCESSING

   DESCRIPTOR--* QUERY CONDITIONS

   DESCRIPTOR--* PREMISE ACTION

   DESCRIPTOR--* ACCESS CONTROL
   DESCRIPTION-* PRIVACY LOCKS BY DATA ELEMENT (AT ANY LEVEL) OR BY FUNC
        TION

   DESCRIPTOR--* DATA ITEM DEFINITION
   DESCRIPTION-* NAME, LENGTH, VALUE SET, UNITS OR DIMENSIONS, EXSISTENC
        E, ACCESS LOCKS, USAGE, ACTUAL OR DERIVED, SINGLE OR MULTIVALUE
        D, AND EDITING RULES
```

ID YYAUTHOR: 71CODASYL

TITLE: "Feature Analysis of Generalized Data Base Management Systems"

Feature Analysis compares ten systems on a number of common dimensions. The ten systems are: GIS, MARK IV, NIPS/FFS, TDMS, UL/1, COBOL, DBTG, IDS, IMS, SC-1. General points made in the report are: (1) the host versus self-contained systems dichotomy is beginning to merge; (2) the distinction between the procedurally oriented programming user interface and the non-procedural non-programming user interface; (3) the importance of data independence and late binding; (4) the distinction between the logical data structure and the physical storage structure used to implement the logical structure; and (5) most existing structures are hierarchical, but the need is for more complex network structures in the future. Data structures are composed of a hierarchy of elements -- item, group, group relation, entry, file, and database.

Considerations in defining the data item include: name, length, value set, units or dimensions, indications of existence or non-existence, access locks, usage of the item, whether it is actual or derived, I/O editing rules, and whether the item is multivalued or not. Considerations for defining groups include: whether it is simple or compound (composed only of item or also of other groups), whether it is repeating and if so the maximum number, whether or not duplicates are allowed in the group and if so where they are inserted (first or last), whether the group members are treated as a set or whether they are ordered on some key and if so how they are ordered, how inserts are done to the group (first, last, next, prior, or ordered based on some key). Privacy locks can be applied in either or both of two ways -- by data element at any level (item, group, file) or by function (for read only, update, insert, or delete).

Interrogation and update is based on a premise:action pair. The premise is a boolean expression involving relationships (equal, not equal, less than, greater than, etc.) or existence and may refer to one or many data items. The actions which can be specified, while they have much in common, tend to be procedural for the programming user and ideally non-procedural for the non-programming user.

ID YYAUTHOR: 71CODASYL
TITLE: "Feature Analysis of Generalized Data Base Management Systems"

Updates are discussed primarily in terms of transaction processing. The transactions may be pre-defined or self-defining (the definition included with the transaction). A distinction is made between changing an item or a record versus inserting or deleting one. Either the transactions, or the master file against which they are processed, or both may be ordered on some key.

```
ID YYAUTHOR-# 71COMPUVISOR
TITLE-------# ASAP FILE MAINTENANCE AND INFORMATION RETRIEVAL SYSTEM RE
    FERENCE MANUAL
PUBLISHER----------# COMPUVISOR, INC.
PUBLISHER LOCATION-# ITHACA, N.Y.
YEAR---------------# 1971
DATE---------------# 1971/09

  AUTHOR NAME-# COMPUVISOR
  FIRST NAME--# INC

  DESCRIPTOR--# UPDATE
  DESCRIPTION-# INCLUDES ACTIVATE AND DEACTIVATE OPERATIONS

  DESCRIPTOR--# ACCESS CONTROL MECHANISMS
  DESCRIPTION-# CONTROLS ACCESS TO DEFINED ITEMS, SYSTEM FUNCTIONS, DEF
         INED PROCESSES, DEFINENEW ITEMS OR RECORDS, DATA DEPENDENT CRIT
         ERIA
  DESCRIPTOR--# ENCODE/DECODE
  DESCRIPTION-# EITHER DIRECTION AS SPECIFIED BY USER, RESIDUAL CODES,
         CODES ON RANGES OF VALUES

  DESCRIPTOR--# TEXT ITEM TYPE
  DESCRIPTION-# WITH CONCATENATION, AND COMPRESSION OPERATORS

  DESCRIPTOR--# COMPRESSION OPERATOR
  DESCRIPTION-# ON TEXT ITEM TYPES

  DESCRIPTOR--# DATABASE OPERATIONS
  DESCRIPTION-# INVERT PARTS OF ITEM AROUND COMMAS(REVERSE), TEST FOR C
         HANGE IN AN ITEMS VALUE DURING A RUN
```

ID YYAUTHOR: 71COMPUVISOR

TITLE: "ASAP File Maintenance and Information Retrieval System Reference Manual"

ASAP is a flat file manager which can handle different record types. Its language is procedural.

Arrays and stacks of one or more items are supported.

Concatenation of text is normal or right or left compress. The latter compress or remove leading blanks from the right operand and trailing blanks of the left operand before concatenation.

Some functions provided are:

reverse --	reverse the values in a field around the commas; useful for a persons name
dollar --	insert a $ and commas in a field
change --	test if the value of a field has changed in this run
bit --	extract a bit from an item

An update process can make a record inactive or active. A function is provided to test if the record is active or inactive.

An encode/decode table is provided with internal value = external value. The right and left side have separate names. A value can be converted by specifying either the right or left table name. The value is converted to the opposite sides value. The word "other" in either side specifies the code for any value not in the table. If one side of the table is numeric and increasing then a greater than or equal (GE) decode/encode can be performed. If the item is GE to the number but less than the next number it is encoded to the value associated with the first number.

Tags specify actions to be performed when a certain event occurs. A tag tests for the event and performs the actions. All tags are executed at the end of a run.

ID YYAUTHOR: 71COMPUVISOR

TITLE: "ASAP File Maintenance and Information Retrieval System Reference Manual"

Security mechanism controls access to the system, access to field and records, and access to system functions. System access is through a user name and password. Associated with this are access to system functions of update, define new items and records, use of defined processes. Each item can have a security class number. Users with the same number can access that item. Also a user can be restricted to certain records (data dependent security) by a boolean expression. This boolean expression is concatenated to any file section expressions the user uses. Default security allows limited access to the file and is very restrictive.

```
ID YYAUTHOR-# 71CONWAY + MAXWELL + MORGAN
TITLE-------# #ON THE IMPLEMENTATION OF SECURITY MEASURES IN INFORMATIO
      N SYSTEMS#
SERIAL TITLE-------# COMMUNICATIONS OF ACM
SERIAL NUMBER------# 15:4
PUBLISHER----------# ACM
YEAR---------------# 1972
DATE---------------# 1972/04
PAGES--------------# 211-220

  AUTHOR NAME-# CONWAY
  FIRST NAME--# R.W.

  AUTHOR NAME-# MAXWELL
  FIRST NAME--# W. L.

  AUTHOR NAME-# MORGAN
  FIRST NAME--# H. L.

  DESCRIPTOR--# ACCESS CONTROL MECHANISMS
  DESCRIPTION-# MATRIX, ROWS ARE USERS, COLUMNS ARE DATA ITEMS, ELEMENT
          IS AUTHORIZATION ACTION

  DESCRIPTOR--# SECURITY MONITORING
  DESCRIPTION-# KEEP A LOG OF UNSUCCESSFUL ACCESS ATTEMPTS

  DESCRIPTOR--# DATA DEPENDENT ACCESS CONTROL
  DESCRIPTION-# MUST BE CHECKED AT RUN TIME
```

ID YYAUTHOR: 71CONWAY + MAXWELL + MORGAN

TITLE: "On the Implementation of Security Measures in Information Systems"

This paper discusses the problem of allowing specific users access to selected portions of the data base.

A security matrix is proposed. The columns are data items -- not necessarily disjoint. The rows are users and an element is a decision rule (read, write, etc.). Most entries are denial of access resulting in a sparse matrix.

The matrix is used as follows:

- a row is selected to authenticate the user
- a subset of columns in this row are selected to determine access to data items
- this subset remains in effect during the entire operation

In "batched" transactions, there is high repetition on the same data items. Security check should take advantage of this for efficient operation. The random inquiry has low repetition and could tolerate higher overhead.

Security monitoring should detect assaults upon the system (repeated attempts to enter system). A log should be kept of successful and unsuccessful access to data and system.

If access is denied to an item (data independent) the security check can be performed at translation time thus avoiding routine overhead.

If access is conditioned (based upon values of the item or other items) then access must be checked at execution time.

```
ID YYAUTHOR-* 71HOFFMAN
TITLE-------* *THE FORMULARY MODEL FOR FLEXIBLE PRIVACY AND ACCESS CONT
     ROLS*
SERIAL TITLE-------* PROCEEDINGS NATIONAL COMPUTER CONFERENCE
SERIAL NUMBER------* VOL 41
PUBLISHER----------* AFIPS
YEAR---------------* 1971
PAGES--------------* 587-601

  AUTHOR NAME-* HOFFMAN
  FIRST NAME--* LANCE J.

  DESCRIPTOR--* ACCESS CONTROL
  DESCRIPTION-* USER WRITTEN AUTHORIZATION ROUTINES WHICH EXECUTE AT RU
       N TIME AND CAN USE DATA DEPENDENT CRITERIA

  DESCRIPTOR--* EXCEPTION CONDITION RESPONSES
  DESCRIPTION-* TO ACCESS CONTROL VIOLATIONS. SYSTEM SHOULD NOT SAY TOO
        MUCH, IF ANYTHING, TO USER ATTEMPTING BREACH OF PRIVACY

  DESCRIPTOR--* ENCRYPTION
  DESCRIPTION-* VERBS TO USE IN A USER WRITTEN ACCESS CONTROL ROUTINE
```

ID YYAUTHOR: 71HOFFMAN

TITLE: "The Formulary Model for Flexible Privacy and Access Controls"

Simply proposes that users (Database Administrators) can provide their own routines and procedures for access control. Provides an extension of the usual table look up and password form of security. Anything the user can program can become part of his access control. Assumes all access control checks will be made at run time. Ignores the fact that data independent (schema level) checks can be made at compile time for improved efficiency. Interesting feature is that it allows different responses when different users get a security violation. Important capability since the response the user gets can be critical. Too much response (e.g., you aren't allowed access to salaries over $10,000) can give the user information he should not have, while too little information (no salaries over $10,000) can introduce errors into his results.

The formulary model uses a User Control Block which identifies the user and links him to the various access control or "formulary" routines. The user goes through a TALK routine to the ACCESS routine which does the actual data STORE and FETCH, depending on the response of the CONTROL routine. The formulary model proposed allows for virtual addressing of the data, encrypting and decrypting of the data, and any type of control procedure to determine if the user will have access to the data. If he is not allowed access, the FETCH and STORE operations in the ACCESS routine are inhibited.

Actually the model is even more general, since anything the user wants to code and pay the price to run can become part of his access control mechanism.

```
ID YYAUTHOR-# 71TONIK
TITLE--------# ≠RECOVERY OF ONLINE DATA BASES≠
SERIAL TITLE--------# PROCEEDINGS ACM NATIONAL CONFERENCE
PUBLISHER-----------# ACM
YEAR----------------# 1971
PAGES---------------# 103-111

   AUTHOR NAME-# TONIK
   FIRST NAME--# AL B.

   DESCRIPTOR--# BACKUP AND RECOVERY
   DESCRIPTION-# THREE CASE STUDIES, TYPES OF ERRORS, BACKUP STRATEGIES,
         AND RECOVERY PROCEDURES

   DESCRIPTOR--# DATA LOSS
   DESCRIPTION-# TYPES OF FAILURES AND EXAMPLES OF BEHAVIORAL DATA ON TH
        OSE FAILURES

   DESCRIPTOR--# BEHAVIORAL CHARACTERISTICS
   DESCRIPTION-# OF DATA LOSSES AND SYSTEM FAILURES
```

ID YYAUTHOR: 71TONIK

TITLE: "Recovery of On-Line Data Bases"

Looks at the back up and recovery procedures for three systems.

1. AMIS -- California Department of Motor Vehicles Driver's License and Motor Vehicle Registration Databases. Approximately 35 million records and 1.3 million urgent queries per month (1971). Non-urgent queries are batched. Four computers are connected to provide the entire system. There is an automatic switch over so that when one fails the system is reconfigured and is back up in 90 seconds. For file recovery they use differential dumps (a quarter of the database is dumped each week). After images are logged for any track that is modified. To recover, reload the last correct after image and reprocess from that point. Six senior operators are regularly assigned to back up/recovery operations.

2. Pillsbury -- is primarily an online retrieval and batch update system with primarily old mature applications. Types of failure: (1) application abort; (2) operating system or hardware failure; (3) disk unreadable; and (4) erroneous update. If recovery is not automatic, visibility is important so that errors can be caught early before they are propogated through the database. For recovery they use daily dumps with overlapping retention (daily, weekly, monthly, annual). The system journal is used for both accounting audit and system recovery. Log includes transactions and before and after images. When application aborts, journal closed, database placed in abort status and inaccessible until recovery. They estimate their roll back procedures three times faster than reload and reprocess. Recovery statistics:

- 6 single job aborts/week
- .5 erroneous updates/week
- 2.5 hardware/software failures/ week affecting one or more data base jobs

ID YYAUTHOR: 71TONIK
TITLE: "Recovery of On-Line Data Bases"

3. Ohio Bell -- an online retrieval and update system with 1200 files and 8000 to 9000 transactions daily with transactions averaging 200 file accesses. The entire data base is dumped nightly and stored off site. Logs are maintained with transactions, before and after images. Four levels of errors: (1) communications errors -- data not accepted; (2) input validation errors -- data not accepted; (3) abort during a transaction -- automatic roll back using before images; and (4) errors discovered after the transaction has completed or not related to a transaction (hardware/software failures) -- manual system restoration required. Three types of level 4 system restorations: (1) short term -- go back to the last time point and write before images of all uncompleted transactions and reprocess; (2) to back to any specified time point (determined manually) write before images and reprocess; (3) long term -- reload and reprocess.

Quantities of errors by type:

level	type	number of errors
1	communications errors	190/day
2	input validation errors	140/day
3	abort during transaction	250/day
4	system restoration (short)	23 in 6 months
4	system restoration (medium)	3 in 6 months
4	system restoration (long)	1 in 2 years

This is one of the few articles that gives any error statistics. Such statistics are important behavioral characteristics which will help determine recovery procedures.

```
ID YYAUTHOR-* 72DANA + PRESSER
TITLE-------* ≠AN INFORMATION STRUCTURE FOR DATA BASE AND DEVICE INDEPE
     NDENT REPORT GENERATION≠
SERIAL TITLE-------* PROCEEDINGS NATIONAL COMPUTER CONFERENCE
SERIAL NUMBER------* VOL 42
PUBLISHER----------* AFIPS
YEAR---------------* 1972
DATE---------------* 1972 FALL
PAGES--------------* 1111-1116

  AUTHOR NAME-* DANA
  FIRST NAME--* C.

  AUTHOR NAME-* PRESSER
  FIRST NAME--* L.

  DESCRIPTOR--* REPORT DEFINITION
  DESCRIPTION-* FOCUSES ON LOGICAL USER SPECIFICATIONS

  DESCRIPTOR--* LOGICAL PAGE SIZE
  DESCRIPTION-* MAPPED TO PHYSICAL DEVICE PAGE SIZE

  DESCRIPTOR--* DEVICE INDEPENDENCE
  DESCRIPTION-* IN REPORT DEFINITION
```

ID YYAUTHOR: 72DANA + PRESSER

TITLE: "An Information Structure for Data Base and Device Independent Report Generation"

Two problems with current output specification: (1) too few defaults for user; and (2) device dependent. Proposes solution in terms of logical (specified by user) and physical (device dependent) report definition and a system mapping between them. If LRU (logical report unit -- page size) greater than PRU, then logical page broken up vertically or horizontally or both. If LRU less than PRU, then pages may be combined. Should be defaults for these mapping which the user could override. Even if PRUs same size, different mappings needed for hard and soft copy. With hard copy could use fold-out pages or look at multiple pages simultaneously.

User specifies: (1) formatting -- page dimensions, header and trailer labels, and row and column spacing (either in absolute or relative terms); (2) actions (input, output, computation, and test) to be taken to generate the report; and (3) specification and definition of the items to be included. Relationship between (3) and the data base definition of the items is not clear.

Article focuses on description of table driven implementation of report generator, rather than on what the user must specify.

```
ID YYAUTHOR-* 72SENKO
TITLE-------* ≠DETAILS OF A SCIENTIFIC APPROACH TO INFORMATION SYSTEMS≠
MONOGRAPH TITLE----* DATA BASE SYSTEMS
EDITOR--------------* RUSTIN, R.
PUBLISHER-----------* PRENTICE-HALL, INC.
PUBLISHER LOCATION-* ENGLEWOOD CLIFFS, N.J.
YEAR----------------* 1972
PAGES---------------* 143-174

   AUTHOR NAME-* SENKO
   FIRST NAME--* M. E.

   DESCRIPTOR--* TRANSACTION DEFINITION

   DESCRIPTOR--* TRANSACTION PROCESSING
   DESCRIPTION-* SPECIFICATION OF FOR SYSTEM DESIGN

   DESCRIPTOR--* DATABASE DESIGN PROCEDURE
   DESCRIPTION-* FOREM AND PHASE II

   DESCRIPTOR--* INFORMATION SYSTEM MODELLING
   DESCRIPTION-* PRIMARILY AT THE PHYSICAL LEVEL OF A DATA STRUCTURE, AC
          CESS ROUTINES, HARDWARE, AND REQUESTS

   DESCRIPTOR--* INFORMATION SYSTEM REQUIREMENTS SPECIFICATIONS
   DESCRIPTION-* IN TERMS OF APPLICATIONS, DATA STRUCTURE, AND HARDWARE
```

ID YYAUTHOR: 72SENKO

TITLE: "Details of A Scientific Approach to Information Systems"

Three strategies to study information systems: (1) look at and analyze "real world" systems; (2) develop taxonomy for "real world" systems; and (3) develop and analyze models of information systems. Actually a sequence of steps, but little progress beyond step two. Two major areas to consider in studies: (1) function -- what should the system do; and (2) efficiency -- how does the system do it and how well. Most work in the area of efficiency. Systems hierarchy:

1. users job -- a set of transactions to be processed
2. application procedures -- what needs to be done (procedure oriented)
3. Information Management System -- data structures and logical access paths
4. operating system
5. physical access methods
6. hardware

For strategy three, need a generalized simulation model which will allow you to define the user's job stream of transactions and procedures (assumption that data access, not actual processing, is the time consuming part of the operations -- business not scientific applications). At the information systems level need to specify data structures and logical access paths. Of the three basic structures (hierarchy, entity set description (similar to relational) and binary relations) the second method was used. This is the only type of structure simulated.

Points made in the article are the underlying rationale for FOREM (and later PHASE II), which simulates systems performance once the user has defined his applications, data structure, and hardware. Parameters which the user must specify for the simulation are a subset of what he must specify for data definition, patterns of processing, and the behavior of both the data and the processes. Also some of the simulation results will be fed back into the design database as improved performance estimates.

```
ID YYAUTHOR-# 72TEICHROEW
TITLE-------# #A SURVEY OF LANGUAGES FOR STATING REQUIREMENTS FOR COMPU
     TER-BASED INFORMATION SYSTEMS#
SERIAL TITLE-------# PROCEEDINGS NATIONAL COMPUTER CONFERENCE
SERIAL NUMBER------# VOL. 40
PUBLISHER----------# AFIPS
YEAR---------------# 1972
PAGES--------------# 1203-1224

  AUTHOR NAME-# TEICHROEW
  FIRST NAME--# DANIEL

  DESCRIPTOR--# INFORMATION SYSTEM REQUIREMENTS SPECIFICATIONS

  DESCRIPTOR--# SYSTEM DEVELOPMENT PHASES
  DESCRIPTION-# ANALYSIS, DESIGN, AND CONSTRUCTION

  DESCRIPTOR--# PSL

  DESCRIPTOR--# INFORMATION SYSTEM REQUIREMENTS STATEMENT LANGUAGE
  DESCRIPTION-# COMPARISON OF SEVERAL INCLUDING ADS, TAG, INFORMATION A
       LGEBRA, LANGEFORS, LOMBARDI, AND YOUNG AND KENT

  DESCRIPTOR--# INFORMATION ALGEBRA

  DESCRIPTOR--# PATTERNS OF PROCESSING
  DESCRIPTION-# HIGH LEVEL LANGUAGE FOR EXPRESSING
```

ID YYAUTHOR: 72TIECHROEW

TITLE: "A Survey of Languages for Stating Requirements for Computer-Based Information Systems"

Focuses on how to specify information systems requirements. Three phases in development -- analysis, design, and construction. Emphasis of paper is on how needs are documented and transferred from the analysis to the design phase. The purpose is to reduce the distance between the person with the problem or ultimate user and the computer.

Must distinguish between Business Data Specification Functions (BDSF) and Data Processing Functions (DPF). The difference is between what to do and how to do it. Problem with general purpose programming languages is that they mix the two types of statements (specifying that a report needs to be ordered on a particular variable is not the same as specifying to sort it). The BDSFs provide a high level, user-oriented set of patterns of processing.

There is a need for a high level requirements statement language to provide a formal method of documentation during the analysis phase. The paper compares a number of such requirements statement techniques, including ADS, TAG, Information Algebra, Systematics, and the works of Young and Kent, Langefors, and Lombardi. The comparisons are on how the problem is defined, how the output is defined, and the data and computational relationships. Few, if any, of these systems are used to any great extent, partly for technical reasons (they aren't satisfactory for stating requirements) and partly for lack of acceptance by the analyst (part of which may be due to all the systems' inadequacy in expressing complete requirements).

Teichroew proposes a set of three objectives for Requirements Statement Languages. First, it must be able to handle both current and future requirements (including changes in hardware, number and relationships among users, more integrated data structures (databases) and less predictable ad hoc uses). Second, the language should be easily used as part of the necessary activities for determining the stating requirements. The language should be usable by the end-user as well as the analyst, allow a top down approach, and aid the user in determining the implications of his requirements on system performance. Third, "the language should be suitable for building the system to accomplish the requirements."

PSL (Problem Statement Language) is an attempt to provide these features.

```
ID YYAUTHOR-# 73COMSHARE
TITLE-------# COMPOSIT 77 USERS GUIDE
PUBLISHER----------# COM SHARE, INC.
PUBLISHER LOCATION-# ANN ARBOR, MI.
YEAR---------------# 1973

  AUTHOR NAME-# COM SHARE
  FIRST NAME--# INC.

  DESCRIPTOR--# DATABASE MANAGEMENT SYSTEM
  DESCRIPTION-# ONLY ACCESSED VIA ONLINE, TIMESHARING FACILITY

  DESCRIPTOR--# DATA STRUCTURE
  DESCRIPTION-# FLAT FILE(SINGLE OR COORDINATED)

  DESCRIPTOR--# DATA ITEM DEFINITION
  DESCRIPTION-# ENCODE-DECODE AND ENCRYPTION

  DESCRIPTOR--# CREATION
  DESCRIPTION-# RULES FOR HANDLING MISSING DATA

  DESCRIPTOR--# FILE LEVEL DERIVATION
  DESCRIPTION-# SUCH AS STATISTICAL OUTPUT, CORRELATION, AND REGRESSION

  DESCRIPTOR--# GRAPHICAL OUTPUT

  DESCRIPTOR--# FORMATION AND PRESENTATION
  DESCRIPTION-# TABULAR, TWO DIMENSIONAL, GRAPHICAL

  DESCRIPTOR--# ENCODE/DECODE

  DESCRIPTOR--# ENCRYPTION
```

ID YYAUTHOR: 73COMSHARE
TITLE: Composit 77 Users Guide

Composit is flat file manager with single fixed length record types. It provides a procedural language with many operations.

Data items are defined with codes for missing values and a scramble option (this item will be scrambled).

A Composit file is created using a data file and a specification file. The specification file describes the format of the data file -- data is in free or fixed field format, record to begin read, number of records to read, input lines per record, an end of record character, a field separator, and the method for handling the file -- a scramble code, treatment of empty (blank) lines (ignore or ignore between records).

Retrieval generates a hit file which can be further processed in a restricted way.

Record selection has multiple AND and OR predicates. Multiple OR selects on an item having many values while multiple AND selects on several items having the same value. A range function and a range function with an increment are provided.

In displaying data the format defaults to columns with item names as titles. Format options include treatment of negative values (CR, minus sign or in parenthesis), treatment of zero values (blank, zero or dash [-]), literals are allowed in the items list, row numbering for output lines.

A prompting option is provided for updating the file. When adding records, the system prints the record number, the item name and asks for its value. When changing values it prints the item name and old value and asks for a new value.

The file can be altered or converted by reordering records, permuting items, adding or removing items, changing the description of an item, changing an items name and its missing values.

ID YYAUTHOR: 73COMSHARE

TITLE: Composit 77 Users Guide

Secondary files, similar in structure to the main file, are supported. These files can be joined (1 to 1) on a common field with mismatches ignored and can be combined using a logical AND or OR on a common item. These files can be saved and later attached for use. Also, the status of files can be printed, item descriptions when file created, when updated and the number of records in the file.

```
ID YYAUTHOR-* 73FOSTER
TITLE-------* #DATA SECURITY DEVELOPMENTS AT IBM#
SERIAL TITLE-------* PROCEEDINGS OF CONFERENCE ON SECURE DATA SHARING
SERIAL NUMBER------* REPORT 4130
PUBLISHER----------* NAVAL SHIP RESEARCH AND DEVELOPMENT CENTER
YEAR---------------* 1973
DATE---------------* 1973/08

  AUTHOR NAME-* FOSTER
  FIRST NAME--* C. LAURENCE

  DESCRIPTOR--* SECURITY
  DESCRIPTION-* ENVIRONMENTS--(1)NATIONAL SECURITY, (2)INTERNAL CORPORA
        TE MIS, AND (3)COMMERCIAL PRIVACY(COMPUTER UTILITY). USER SPECI
        FIES LEVEL OF SECURITY OR CONSEQUENCES IF DATA COMPROMISED.
```

ID YYAUTHOR: 73FOSTER

TITLE: "Data Security Developments at IBM"

Must have an operational definition of security before real progress. IBM working with three installations (TRW, MIT, and State of Illinois Management Division) to study three problems: (1) technology involved in security; (2) to install a system which is "relatively" secure, although not completely so; and (3) to evaluate that system against the technology. Work has identified three types of environments: (1) those involving defense and national security; (2) those involving internal corporate management information and product development; and (3) the commercial privacy environment (unclear whether this includes both large commercial credit or insurance databases with a limited number of user and computer utilities). Certification is discussed with respect to certifying a standard level of performance (analogy made to 30 minute, one hour and ten hour safes as being secure in terms of fixed criteria). Must also consider economic trade off between cost of security and cost if data compromised.

As part of his problem definition the user must define the level of security required by his procedures and data. However, he should be unconcerned how those requirements are implemented, except to insure that those requirements are met (this is the need for certification). Another approach would be for the user to provide some indication of the consequences if his data is compromised and allow the system to select the appropriate level of security from its various alternatives.

```
ID YYAUTHOR-* 73KING + COLLMEYER
TITLE-------* ≠DATABASE SHARING -- AN EFFICIENT MECHANISM FOR SUPPORTIN
    G CONCURRENT PROCESSES
SERIAL TITLE-------* PROCEEDINGS NATIONAL COMPUTER CONFERENCE
SERIAL NUMBER------* VOL 43
PUBLISHER----------* AFIPS
YEAR---------------* 1973
PAGES--------------* 271-275

   AUTHOR NAME-* KING
   FIRST NAME--* PAUL F.

   AUTHOR NAME-* COLLMEYER
   FIRST NAME--* ARTHUR J.

   DESCRIPTOR--* CONCURRENT PROCESSES

   DESCRIPTOR--* UPDATE PROCESSES
   DESCRIPTION-* CONCURRENCY CONTROL

   DESCRIPTOR--* LOCKS ON DATA
   DESCRIPTION-* FOR CONCURRENT PROCESSES

   DESCRIPTOR--* DATA DEPENDENT UPDATE
   DESCRIPTION-* USER RESPONSIBILITY IN CONCURRENT ACCESS

   DESCRIPTOR--* CONCURRENCY USER COMMANDS
   DESCRIPTION-* LOCK/UNLOCK PROCESS, ALLOCATE/DEALLOCATE DATA

   DESCRIPTOR--* DEADLOCK
   DESCRIPTION-* PREVENTION

   DESCRIPTOR--* RECOVERY
```

ID YYAUTHOR: 73KING + COLLMEYER

TITLE: "Database Sharing -- An Efficient Mechansim for Supporting Concurrent Processes"

Key to database sharing is simultaneous access. Limits are required to prevent concurrent updates, but they create potential for deadlock. DBTG's use of KEEP-FREE avoids deadlock, but places integrity burden on user -- dangerous approach. Requirements for data driven applications too unpredictable for usual deadlock prevention methods.

Proposes two mechanisms. LOCK/UNLOCK -- used by process to indicate it will want sole access. ALLOCATE/DEALLOCATE -- used to identify the actual data items needed. LOCK creates a "lock list" for a process. ALLOCATE/DEALLOCATE attaches and removes items from the lock list. Deadlock (as opposed to a temporary block) occurs when the data base state graph (represented by all lock lists) contains a loop. Tracing the lock lists will identify both the processes which are deadlocked and the item over which they are deadlocked.

Recovery involves restarting a process at its last checkpoint, which was automatically taken when the LOCK was issued. However, data base does not need to be recovered because it is not changed until DEALLOCATE. Can restart either process, depending on system criteria (e.g., shortest lock list, lowest priority, etc.).

Authors undecided whether LOCK and ALLOCATE should be explicit command or implicit in part of other commands. Important that read only process should also have option to LOCK and ALLOCATE. Mechanisms only mandatory for data dependent updates.

```
ID YYAUTHOR-* 73NUNAMAKER + SWENSON + WHINSTON
TITLE-------* ≠SPECIFICATIONS FOR THE DEVELOPMENT OF A GENERALIZED DATA
      BASE PLANNING SYSTEM≠
SERIAL TITLE-------* PROCEEDINGS NATIONAL COMPUTER CONFERENCE
SERIAL NUMBER------* VOL 43
PUBLISHER----------* AFIPS
YEAR---------------* 1973
PAGES--------------* 259-270

  AUTHOR NAME-* NUNAMAKER
  FIRST NAME--* J. F.

  AUTHOR NAME-* SWENSON
  FIRST NAME--* D. E.

  AUTHOR NAME-* WHINSTON
  FIRST NAME--* A. B.

  DESCRIPTOR--* PROGRAM GENERATION
  DESCRIPTION-* IN RESPONSE TO INQUIRY OR REQUEST

  DESCRIPTOR--* APPLICATION SYSTEM
  DESCRIPTION-* AS IT RELATES TO DMS
```

ID YYAUTHOR: 73NUNAMAKER + SWENSON + WHINSTON

TITLE: "Specifications for the Development of a Generalized Data Base Planning System"

Wants a system which as a result of a query can: (1) retrieve the data needed to answer the query; and/or (2) set up the application program or model that must be run to answer the query. Data base systems ease the first step, but users also need the second step. GPLAN (Generalized Database Planning System) proposed as such a system. It is a synthesis of four components: (1) NAPSS -- Numerical Analysis Problem Solving System; (2) SODA -- Systems Optimization and Design Algorithm; (3) GDMS -- Generalized Data Management System; and (4) OPTIMA -- a mathematical programming and optimization system. NAPSS and SODA are similar in that they both have a number of algorithms and automatically select the best one for the application. Under GDMS he discusses four systems: System/2000, RAMIS, IMS, and DISK FORTE.

Lists nine components of a Generalized Database Planning System: (1) a GDMS to allow input, search, storage, maintenance, retrieval, and output; (2) the actual data in the database; (3) a query language which will allow queries against the data base and provide directives to the application packages and models; (4) an analyzer for the query language; (5) a collection of application packages and models; (6) an administrative report module; (7) user interface; (8) a set of extraction files as intermediaries between the database and the application packages (definition of the content and structure of the files and how they are used by the various applications); and (9) the users (two types -- (a) the technical computer systems oriented personnel and (b) the non-technical administrator or manager who has the planning application).

This is one of the few articles that addresses both the database and the applications side of the entire problem -- solving the user's problem. Because of the complexity of the analysis and results there is also an emphasis on the use of graphic rather than tabular output. In summary, the GPLAN attempts to do for planning what ISDOS tries to do for computer systems design.

```
IC YYAUTHOR-# 74CCMPUTER CORPORATION OF AMERICA
TITLE-------# CCA204 USER LANGUAGE REFERENCE MANUAL
PLBLISHER----------# COMPUTER CORPORATION OF AMERICA
PLBLISHER LCCATION-# CAMBRIDGE, MA
YEAR---------------# 1974
CATE---------------# 1974/03

  AUTHOR NAME-# CCMPUTER CORPORATION OF AMERICA

  DESCRIPTCR--# DATABASE MANAGEMENT SYSTEM

  DESCRIPTCR--# DATA STRUCTURE
  DESCRIPTICN-# SINGLE FLAT FILE(INVERTED), LATER RELEASE FOR COORDINAT
     ED FILES. NOW MULTIPLE FLAT FILES ONE AT A TIME

  DESCRIPTCR--# ENCODE/DECODE
  DESCRIPTICN-# VARIABLE ITEMS ALWAYS ENCODED

  DESCRIPTCR--# REPORT DEFINITION

  DESCRIPTCR--# AUDIT
  DESCRIPTICN-# PRINTS USER SPECIFIED AUDIT TRAIL

  DESCRIPTCR--# CONCURRENT CONTROL
  DESCRIPTICN-# LOCKS AT FILE LEVEL FOR MORE THAN ONE UPDATE

  DESCRIPTCR--# UPDATE PROCESSES
  DESCRIPTICN-# OPERATORS TO CHANGE, ADD OR DELETE FIELDS OR RECORDS

  DESCRIPTCR--# SELECTION CRITERIA
  DESCRIPTICN-# ON ITEM VALUES OR RANGES. MULTIVALUED ITEM DEFAULT ONLY
      FIRST INSTANCE, OTHERWISE ANY INSTANCE
```

ID YYAUTHOR: 74COMPUTER CORPORATION OF AMERICA
TITLE: CCA204 User Language Reference Manual

Model 204 is a complete (self-contained and hosted) retrieval and update system which can be used in a batch or online environment. It uses a single flat file structure with limited repeating group or multivalued item capability. Although a Model 204 database can contain up to 255 files, only one can be opened at any time. It uses IFAM (Inverted File Access Method) and allows four logical access paths; inverted, direct, sequential, and linked or chained.

The user has only logical data definition capabilities; name, type, and if the field is repeating the maximum number of occurrences. The type is a composite of several categories:

- key (default) or non-key
- invisible -- user specified pointer which cannot be printed
- numeric -- character, binary, or coded
- date
- suppress FRV (for each value) -- used when values are almost unique

Names are automatically encoded, as are values, when their lengths are variable.

Records are selected based on item(s), value(s), or range(s). For repeating or multivalued items the default is to consider only the first item, but there is an option to let the selection be on ANY of the item values. The system will return either the selected set of records or a list of pointers. The list can be concatenated with or processed against other lists. Selection can only be on keyed items.

Updates to the database can include value modification of items or adds or deletes of either items or records. For multivalued items adds are done to the end of the set. For value modification the default is to change only the first item, but this can be overridden by specifying change any item with valuel to value2. This form of change can also be used with non-repeating items. Similarly, you can delete all of the repeating items or only those with a certain value. For update protection the system locks at the file level. Requests are classified as update, retrieve only, or non-file. During an update, retrievals and other updates are locked out. There is also an AUDIT command which prints out an "audit trail" with a set of user specified items.

The user can define and save his own processes or SEGMENTs which he can later invoke with a set of parameters. The user can output items in a default format by just giving its name. However, there are also extensive report definition capabilities in the system.

```
ID YYAUTHOR-# 74HAMMER
TITLE-------# ≠DATA ABSTRACTIONS FOR DATA BASES≠
SERIAL TITLE-------# SIGPLAN NOTICES
SERIAL NUMBER------# 8:2
MONOGRAPH TITLE----# PROCEEDINGS CONFERENCE ON DATA ABSTRACTION, DEFENI
      TION AND STRUCTURE
PUBLISHER----------# ACM
YEAR---------------# 1976
PAGES--------------# 58-59

    AUTHOR NAME-# HAMMER
    FIRST NAME--# MICHAEL J.

    DESCRIPTOR--# DATABASE OPERATORS
    DESCRIPTION-# HIGH LEVEL, APPLICATION ORIENTED

    DESCRIPTOR--# ABSTRACT DATA TYPE
    DESCRIPTION-# DATA STRUCTURE PLUS ALLOWED OPERATORS
```

ID YYAUTHOR: 76HAMMER
TITLE: "Data Abstractions for Data Bases"

The premise is that a user should view the data base in behavioral terms. He should be concerned with the behavioral semantics of data and not so much with its structure. This approach will provide a greater degree of data independence.

By limiting access to the data base to a few semanticly meaningful operations the user is isolated from the data structure. In an employee data base they may be HIRE, FIRE, GIVE RAISE, etc.

This concept also protects the data base by limiting access to data and preserving its integrity.

The problems of this approach are:

- a database is defined in terms of data not in terms of operations
- new operations develop after the data base has been defined

Issues for resolution:

- not all interactions to the data base can be defined as abstract operations
- a users view and use of the database changes over time -- the operations must evolve
- different users may define different operations on the same schema (subschema). It must be assured that these operations are consistent with other subschemas and that they do not interfere with each other.

```
IC YYAUTHOR-* 74LEAVENWORTH + SAMMET
TITLE-------* ≠AN OVERVIEW OF NON PROCEDURAL LANGUAGES≠
SERIAL NUMBER------* 9:4
PUBLISHER----------* ACM
YEAR---------------* 1974
DATE---------------* 1974/03
PAGES--------------* 1-12

   AUTHOR NAME-* LEAVENWORTH
   FIRST NAME--* BURT M.

   AUTHOR NAME-* SAMMET
   FIRST NAME--* JEAN E.

   DESCRIPTOR--* HIGH LEVEL LANGUAGE
   DESCRIPTION-* PROPERTIES OF, INCLUDING ELIMINATION OF ARBITRARY SEQUE
         NCES AND USE OF AGGREGATE OPERATORS ON WHOLE DATA STRUCTURE

   DESCRIPTOR--* DATABASE OPERATORS
   DESCRIPTION-* IN A HIGH LEVEL LANGUAGE

   DESCRIPTOR--* PATTERNS OF PROCESSING
   DESCRIPTION-* FUNCTIONS IN INFORMATION SYSTEM PERFORMED BASED UPON A
         PATTERN IN THE DATA
```

AD-A038 400   MINNESOTA UNIV MINNEAPOLIS MANAGEMENT INFORMATION SY--ETC F/G 9/2
AN AUTOMATED ANNOTATED BIBLIOGRAPHY ON THE SPECIFICATION OF INF--ETC(U)
OCT 76 G C EVEREST, O BRAY, I VALTERS   N00167-76-M-8441

UNCLASSIFIED   MISRC-TR-77-01   NL

2 OF 2
AD
A038400

END
DATE
FILMED
5-77

ID YYAUTHOR: 74LEAVENWORTH + SAMMET
TITLE: "An Overview of Non-Procedural Languages"

This article gives some characteristics or features of very high level languages.

A very high level language (VHLL) is a prescription for solving a problem without sequence of statements.

It is relative to its predecessors. The definition of VHLL changes depending upon what previous languages have done.

VHLL can contain both high level and low level features.

1. Associative referencing: data is accessed based upon its properties (content, etc.).

2. Aggregate operators: operators upon the whole data structure rather than an item or a record at a time.

3. Elimination of arbitrary sequence: for a specific problem the order of statements is irrelevant. Elimination of assignment statements, GO TO's, and temporary variables.

4. Nondeterministic and parallelism: a free choice of statements with multiple paths in a statement(s).

5. Pattern directed: perform functions based upon a pattern in the data -- similar to associative referencing.

```
ID YYAUTHOR-* 74LINDGREEN
TITLE-------* #BASIC OPERATIONS ON INFORMATION AS A BASIS FOR DATA BASE
     DESIGN#
PUBLISHER----------* IFIP
YEAR---------------* 1974
PAGES--------------* 993-997

  AUTHOR NAME-* LINDGREEN
  FIRST NAME--* PAUL

  DESCRIPTOR--* DATABASE DESIGN
  DESCRIPTION-* DECIDING DATA TO BE INCLUDED AND OPERATIONS TO PERFORM
       ON DATA
  DESCRIPTOR--* INFORMATION ALGEBRA

  DESCRIPTOR--* DATA STRUCTURE
  DESCRIPTION-* FLAT FILE(SINGLE OR MULTIPLE)

  DESCRIPTOR--* ATTRIBUTES
  DESCRIPTION-* MAY DESCRIBE, IDENTIFY, OR RELATE ENTITIES. DOMAIN OF V
       ALUES FOR.
  DESCRIPTOR--* DATABASE OPERATORS
  DESCRIPTION-* INCLUSION,EXTRACTION, CONGLOMERATION, SEPARATION, DERIV
       ATION, DETECTION, AND EXPOSITION--ALL WELL DEFINED IN TERMS OF
       FLAT FILES.
```

ID YYAUTHOR: 74LINDGREEN

TITLE: "Basic Operations on Information as a Basis for Data Base Design"

Basic goal is to aid analysts in moving away from implementation oriented tools toward a more information theoretic approach -- focusing more on what needs to be done rather than on how to do it. From the user's point of view database design consists of deciding what data to include in the database and what operations to perform on the data to produce meaningful information.

Real world is composed of entities, attributes, and values. Attributes may describe, identify, or relate entities. Describing and relating attributes have a one to many relationship with entities, whereas identifying attributes are normally one to one. A domain defines the formal value set of an attribute, although many attributes may share the same domain.

Out of the total attribute space only a certain subset of attributes apply to a particular entity or entity type. This defines the entity's relevant attribute space. An information element is composed of an entity, an attribute, and a value. Information elements are grouped into information sets. Three types of information sets: (1) isotypic - set of entity, attribute, value triplets where the entities have the same attribute, although not necessarily the same value for that attribute; (2) isonymous - the set refers to the attribute value pairs for the same entity; and (3) homogenous - is a flat file containing specified set of attribute value pairs for a given set of entities.

The concepts are best pictured as a matrix where the rows are entities, the columns are attributes, and each cell contains a value. Isotypic select is of a column. Isonymous selection is of a row. A homogenous information set is a combination.

Lindgreen defines seven operators -- six binary set operators and one monadic operator.

1. Inclusion combines two Homogenous Information Sets (HIS). It operates on flat files and simply appends records of the same type.

ID YYAUTHOR: 74LINDGREEN

TITLE: "Basic Operations on Information as a Basis for Data Base Design"

2. Extraction isolates one or more isonymous sets (since isonymous sets refer to the same entity this operator in effect selects one or more entities). It operates on a flat file and selects records.
3. Conglomeration operates on multiple flat files to combine records for the same entity. In effect it concatenates multiple segments of a record. It is similar to inclusion, except that it operates on attributes instead of entities. In terms of the matrix -- inclusion combines rows into a series of records or a file, whereas conglomeration combines columns into a single records.
4. Separation is similar to extraction except that it operates on attributes (columns) instead of entities (rows). It is analogous to the projection relational operator.
5. Derivation defines the value of an attribute in terms of one or more other attributes. It yields an isotypic set where the entities have a common attribute. This operator in effect provides the derivation rule for a derived attribute.
6. Detection is similar to derivation except that it uses a boolean selection expression. While derivation tells how to calculate a value, detection asks is the value equal to a specified value.
7. Exposition is a many to one operator. It is a file level derivation operator, e.g., a summation.

Since all information processing cannot be described completely by this set of operators it is still incomplete, but it is a beginning for the approach Lindgreen favors.

```
IC YYAUTHOR-* 74LISKOV + ZILLES
TITLE-------* #PROGRAMMING WITH ABSTRACT DATA TYPES#
SERIAL TITLE-------* SIGPLAN NOTICES, PROCEEDINGS SYMPOSIUM ON VERY HIG
     H LEVEL LANGUAGES
SERIAL NUMBER------* 9:4
PLBLISHER----------* ACM
YEAR---------------* 1974
CATE---------------* 1974/03
PAGES--------------* 50-59

  AUTHOR NAME-* LISKOV
  FIRST NAME--* BARBARA

  AUTHOR NAME-* ZILLES
  FIRST NAME--* STEPHEN

  DESCRIPTOR--* ABSTRACT DATA TYPE
  DESCRIPTION-* LOGICAL DATA TYPE OR STRUCTURE CHARACTERIZED BY THE OPE
       RATIONS AVAILABLE ON IT

  DESCRIPTOR--* DATABASE OPERATORS
  DESCRIPTION-* ASSOCIATED WITH A DATA TYPE
```

ID YYAUTHOR: 74LISKOV + ZILLES

TITLE: "Programming with Abstract Data Types"

An abstract data type is defined as a class of objects which is characterized by the operations available on it. It is a mechanism for defining data and the legal operations on the data. The data can only be accessed by invoking the operations defined.

The term abstract means the user need not know the physical structure of that data or the way the operations on the data are implemented. He need only know the logical conceptual data structure and what the operations do to this logical structure.

An operational cluster (similar to the CLASS structure in simula) is defined as a TYPE. It consists of local data structures and operations on these local data structures. There is an operation to initialize the local data.

A variable is declared to be of this cluster type. The only way to access and manipulate data of this type is through the operations defined in the cluster.

```
ID YYAUTHOR-# 74MINSKY
TITLE-------# ≠ON INTERACTION WITH DATA BASES≠
SERIAL TITLE-------# PROCEEDINGS SIGMOD WORKSHOP
EDITOR-------------# RUSTIN, RANDALL
PUBLISHER----------# ACM
YEAR---------------# 1974
DATE---------------# 1974/05
PAGES--------------# 51-62
   AUTHOR NAME-# MINSKY
   FIRST NAME--# N.

   DESCRIPTOR--# DATA CONSISTENCY RULES

   DESCRIPTOR--# VALIDATION CRITERIA
   DESCRIPTION-# DATABASE

   DESCRIPTOR--# DATABASE OPERATORS
   DESCRIPTION-# PERMITTED USE

   DESCRIPTOR--# ACCESS CONTROL
   DESCRIPTION-# DATA DEPENDENT

   DESCRIPTOR--# DATA DEPENDENT ACCESS CONTROL
   DESCRIPTION-# AT INSTANCE LEVEL; SIMILAR TO CLASS IN SIMULA

   DESCRIPTOR--# PATTERNS OF PROCESSING
   DESCRIPTION-# PRIMITIVE OPERATIONS

   DESCRIPTOR--# EXCEPTION CONDITION ACTIONS
   DESCRIPTION-# TAKEN WHEN CONSTRAINTS OR PREMISES FAIL
```

ID YYAUTHOR: 74MINSKY

TITLE: "On Interaction with Data Bases"

Data bases, as opposed to files, have an intrinsic meaning independent of their users. These are the formal specifications of data base properties (D-rules). Integrity rules specify domain and consistency constraints and exception actions to be taken when constraints/premises fail. Privacy rules determine what users can do to the data base and what information they can get.

Two ways to specify and insure integrity: (1) functional specification -- define premises and test whenever an item is modified; or (2) constructive specification -- define a set of "legal" operators or primitives which guarantee valid results. Minsky favors constructive specification because: (1) it avoids performance penalties from constantly testing premises; and (2) it solves the problem of allowing the data base to move through temporarily invalid states while processing a transaction. Primitives for data bases still need to be defined since the ones for files (update and retrieve) are not appropriate.

For security, access control is not enough. Need control at both the schema and instance level. For example, user may be allowed an item only if it has certain values. Minsky links this level of control to SIMULA's class.

Much of Minsky's objection to functional integrity based on performance implications. He assumes must always test premise, but part of definition may specify when to test (see Eswaran and Chamberlin). If his primitives were developed, they might provide the basis for the patterns of processing.

```
ID YYAUTHOR-* 74OLLE
TITLE-------* *SELF CONTAINED FACILITIES TO PROCESS SUB STRUCTURES OF M
    ORE COMPLEX NETWORK STRUCTURES*
SERIAL TITLE-------* PROCEEDINGS INTERNATIONAL COMPUTING SYMPOSIUM
SERIAL NUMBER------* 1973
EDITOR-------------* GUNTHER, A, ET. AL.
PUBLISHER----------* NORTH HOLLAND PUBLISHING CO
YEAR---------------* 1974
PAGES--------------* 527-532

   AUTHOR NAME-* OLLE
   FIRST NAME--* T. WILLIAM

   DESCRIPTOR--* HIGH LEVEL MULTIFILE LANGUAGE
   DESCRIPTION-* DESIGN CONSIDERATIONS FOR NON-PROGRAMMING USER INTERFAC
      E

   DESCRIPTOR--* TRADEOFFS
   DESCRIPTION-* MADE BY USER OR SYSTEM. USER KNOWLEDGE FOR TRADEOFF CAN
       BE MANDATORY, OPTIONAL, OR HIDDEN.

   DESCRIPTOR--* SELECTION CRITERIA
   DESCRIPTION-* SHOULD BE SAME IN UPDATE AND QUERY

   DESCRIPTOR--* CONDITIONS ON A PROCESS
   DESCRIPTION-* PREMISE ACTION SEPCIFICATION
```

ID YYAUTHOR: 74OLLE

TITLE: "Self-Contained Facilities to Process Substructures of More Complex Network Structures"

Olle gives some design considerations for interfacing a nonprogrammer to a database. He does this in the light of what the user must know before interfacing to the data base. If a syntax is simple the semantics could be complex. A trade off is considered in terms of simple syntax but complex semantics. This implies more knowledge by the user.

The major points are:

1. The nonprogrammer should not be required to control processing time trade-off. This requires more knowledge about physical and logical structure.
2. If user has a high degree of control, he needs more knowledge and system becomes complex to use.
3. premise/action mix

   limit to <action> <premise> or
   <action> ... <action> <premise>
   one or many actions on a single
   subset of the data base (complex
   syntax)
4. The more complex a data structure (hierarchy, network) that is processed, the more knowledge is required and the more complex the syntax.
5. Selection facilities should be same in update and query. Implies a consistent syntax across functions requiring less knowledge to use.

Optimal language design is with a simple consistent syntax (easier to learn and use) but with easy to understand semantics (function takes few paths).

Cognizance factors:

1. mandatory -- user must know
2. optional -- user may know
3. hidden -- user shouldn't know

```
ID YYAUTHOR-* 74PROGRAM PRODUCTS
TITLE-------* THE DATA ANALYZER TECHNICAL REFERENCE MANUAL
PUBLISHER-----------* PROGRAM PRODUCTS, INC.
PUBLISHER LOCATION-* NANUET, N. Y.
DATE----------------* 1974

  AUTHOR NAME-* PROGRAM PRODUCTS
  FIRST NAME--* INC.

  DESCRIPTOR--* DATABASE MANAGEMENT SYSTEM

  DESCRIPTOR--* DATA STRUCTURE
  DESCRIPTION-* SINGLE OR COORDINATED FLAT FILES(LIMITED USE OF REPEATI
       NG GROUPS --ONLY FOR SELECTION)

  DESCRIPTOR--* DATA DEFINITION
  DESCRIPTION-* MAINLY PHYSICAL(DISPLACEMENT, LENGTH, TYPE). TEMPORARY
       VARIABLES FOR CALCULATION OR OUTPUT

  DESCRIPTOR--* REPORT DEFINITION
  DESCRIPTION-* DEFAULTS OR USER OVERRIDE

  DESCRIPTOR--* QUERY TYPES

  DESCRIPTOR--* PROCESS DEFINITION
  DESCRIPTION-* SEQUENTIAL, PROCEDURAL, FORMS DRIVEN

  DESCRIPTOR--* EXCEPTION CONDITIONS
  DESCRIPTION-* MISMATCH OPTION WITH COORDINATED FILES FOR MISSING MAST
       ER OR SLAVE RECORD
```

ID YYAUTHOR: 74PROGRAM PRODUCTS

TITLE: The Data Analyzer Technical Reference Manual

Data Analyzer is a well-organized, self-contained retrieval system for processing a single flat file or up to six coordinated flat files. It can include either fixed or variable length records. It is a form driven system with separate sections for:

1. application (names the file)
2. report title
3. record selection
4. sort and control totals
5. computations/CALLS/optionsl features (including entry and file level derivations)
6. print specifications (extraction and formatting specifications if default options not used)

Data definition defines the physical layout of the records. Limited use of four character repeating group or multi-valued items which can be used only for selection. The user specifies the name, fixed starting position, type (character, packed decimal, or binary), output default column heading, and value editing information. If there are multi-valued items, the user also specifies the maximum number of occurrences per record.

The basic pattern of processing is for the user to select a subset of records and process them. Selection can be based on one or many data items and can look for a specific value or a range of values. For multi-valued items all items are tested for the condition and if any are found the record is selected. Given a set of records the user can specify the sequence for them. Also he can specify desired control totals and do special processing on any control breaks. Within Data Analyzer the user has limited computational ability for initialization, detail processing, or control break processing. However, he can make CALLS to other catalogued system or user defined processes. The user can create temporary items for operations or output. When processing coordinated files, the user can specify any of several MISMATCH options based on the absence of either the control or slave record. The options are to by-pass the mismatched record, abort the job, or process the record using a dummy for the missing record. The user can also specify the maximum number of

ID YYAUTHOR: 74PROGRAM PRODUCTS
TITLE: The Data Analyzer Technical Reference Manual

records to be searched or selected. For output, default column headings and editing is included in the data definition. However, Data Analyzer also has extensive report definition capabilities which the user can use to override the default definitions. There can be table look up for output value editing. Output can also be edited for decimal point, comma insertion, floating $, and the - or CR notation for negative numbers.

```
ID YYAUTHOR-# 74RANDALL
TITLE-------# #INTERUGATING DATE-SENSITIVE FILES#
SERIAL TITLE-------# THE COMPUTER JOURNAL
SERIAL NUMBER------# 17:4
YEAR---------------# 1974
DATE---------------# 1974/11
PAGES--------------# 302-305

   AUTHOR NAME-# RANDALL
   FIRST NAME--# F. E.

   DESCRIPTOR--# DATE SENSITIVE FILE
   DESCRIPTION-# RECORDS CHANGES OVER TIME, NOT JUST SNAPSHOT OF CURRENT
        STATE

   DESCRIPTOR--# BACKUP DATA
   DESCRIPTION-# USED TO KEEP DATA FOR PAST TIME PERIODS

   DESCRIPTOR--# TIME STAMPING
   DESCRIPTION-# ON STORED DATA ITEMS IN RECORDS AND REPEATING GROUPS AN
        D USED IN SELECTION CRITERIA. EFFECTIVE DATE-WHEN CHANGE OCCURS
        IN REAL WORLD, NOT WHEN DATA ENTERED

   DESCRIPTOR--# SELECTION CRITERIA
   DESCRIPTION-# BASED ON TIME POINT(AT, BEFORE, AFTER, EVER) OR PERIOD(
        IN, DURING, UNTIL, SINCE, ALWAYS)
```

ID YYAUTHOR: 74RANDALL

TITLE: "Interrogating Date-Sensitive Files"

Describes the unique features of a date-sensitive system PRISM (Personnel Record Information System for Management) and its retrieval system PIRL (PRISM Information Retrieval Language). The usual way systems handle time is to keep a current copy of the database and take periodic snapshots for backup and recovery. In these cases the date relates to system entry, not an effective date for the real world application. This type of historical data is not available to the user. PRISM solves this problem by also time stamping the effective date in terms of the date at which the change or addition becomes effective for the application. PRIL then allows retrieval using these time qualifications.

Data definition is not discussed except to mention that there are two types of attributes with respect to changes over time. Some attributes, such as marital status, have only a single current value. When it is changed, a new current value and its time stamp become effective, although the previous value is still available for retrieval with this system. Other attributes, such as skill can be multivalued. When a person acquires a new skill, he does not necessarily lose his old skill. However, there must also be a way to replace or delete a multivalued item when necessary.

Selection expressions in PIRL include the usual boolean expressions with respect to item values or value sets (equal, not equal, less than, less than or equal, etc.). In addition, there are time qualifications which allow specification of either a point in time (AT) or an interval. Intervals can be specified so that selection will be made if a condition was met at least once during the period (IN a period, BEFORE a date, AFTER a date, or EVER). Complementary expressions (DURING, UNTIL, SINCE, and ALWAYS) require that the condition be met at all times during the period. Dates of occurrences can also be pin-pointed with FIRST, LAST, NEXT, PREVIOUS, EARLIER, and LATER.

Output can be either part or all of selected records or statistical tables. With time qualification, formatting the tables can become very complex, but the article doesn't give any details about how it is handled.

ID YYAUTHOR: 74RANDALL

TITLE: "Interrogating Date-Sensitive Files"

This system assumes all data items require time stamping for effective date, but it should be a user option. Conceivably there are four types of items -- single or multi-valued items each with or without time stamping. Also there is a question about the time stamp for a group of items. If each item had a different effective date, what should be the group date. Also must be able to handle errors which should have no effective date. Consider an erroneous value that is entered and dated. When it is corrected, no query should probably find the incorrect value regardless of the data type.

```
ID YYAUTHOR-# 74SAYANI
TITLE-------# ≠RESTART AND RECOVERY IN A TRANSACTION-ORIENTED INFORMATI
    ON PROCESSING SYSTEM≠
SERIAL TITLE-------# PROCEEDINGS SIGMOD WORKSHOP
EDITOR-------------# RUSTIN, RANDALL
PUBLISHER----------# ACM
YEAR---------------# 1974
DATE---------------# 1974/05
PAGES--------------# 351-366

  AUTHOR NAME-# SAYANI
  FIRST NAME--# HASAN

  DESCRIPTOR--# BACKUP AND RECOVERY
  DESCRIPTION-# CLASSIFIES TYPES OF DATA LOSS

  DESCRIPTOR--# RECOVERY PROCEDURES
  DESCRIPTION-# FOR EACH TYPE OF DATA LOSS

  DESCRIPTOR--# DATA LOSS
  DESCRIPTION-# CAUSES: HARDWARE (DATABASE CRASH), SOFTWARE, USER UPDAT
        E PROCESS

  DESCRIPTOR--# BACKUP STRATEGY

  DESCRIPTOR--# RESTART
```

ID YYAUTHOR: 74SAYANI

TITLE: "Restart and Recovery in a Transaction-Oriented Information Processing System"

Discusses recovery policy in a large transaction oriented data base. Identifies three types of errors, eight procedures which allow recovery from them, and seven actions which are the basis of the various procedures. Some actions are event triggered, while others are controlled and done on demand or on a set schedule. The restart and recovery policy is: (1) the set of procedures for recovering from each type of error, (2) the set of actions which are needed for each recovery procedure; and (3) the frequency of the controlled actions.

Errors classified by: (1) severity -- transient, solid, or disaster; (2) cause/origin -- hardware, software, or user; and (3) recovery procedure needed. Examples of errors used are: (1) data base crash -- loss of the entire database; (2) user errors -- incorrect updates entered and propogated; (3) core failure -- part of the data base is lost. Sayani states that the relative frequencies of these failures are 1, 20, and 5.

Possible recovery procedures for each error type are:

A. Data base crash: (1) copy duplicate data base; (2) copy last dump and reconstruct using after images; or (3) copy last dump and reprocess;

B. User error: (4) copy last dump and reconstruct using after images up to the error transactions, then reprocess them and succeeding transactions; or (5) roll back the data base until the initial error and reprocess;

C. Core failure: (6) roll back data base until last checkpoint and reprocess transactions; (7) copy last dump and reconstruct using after images until last complete transaction and then reprocess; or (8) copy last dump and reprocess.

ID YYAUTHOR: 74SAYANI
TITLE: "Restart and Recovery in a Transaction-Oriented Information Processing System"

Each of these procedures requires that certain prior actions were taken. The following actions in various combinations are the basis of all the recovery procedures: (1) dump the data base; (2) log before images; (3) log after images; (4) log transactions; (5) log output messages; (6) take checkpoints, and (7) maintain duplicate database.

Objective of restart and recovery policy (procedures, actions, and frequency of controlled actions) is to minimize restart and recovery cost.

```
ID YYAUTHOR-* 74SHNEIDERMAN + SCHEUERMANN
TITLE-------* #STRUCTURED DATA STRUCTURES#
SERIAL TITLE-------* COMMUNICATIONS OF ACM
SERIAL NUMBER------* 17:10
PUBLISHER----------* ACM
YEAR---------------* 1974
DATE---------------* 1974/10
PAGES--------------* 566-574

 AUTHOR NAME-* SHNEIDERMAN
 FIRST NAME--* BEN

 AUTHOR NAME-* SCHEUERMANN
 FIRST NAME--* PETER

 DESCRIPTOR--* DATA STRUCTURE DEFINITION
 DESCRIPTION-* DEFINED ALONG WITH THE SET OF MEANINGFUL OPERATORS ON T
    HE STRUCTURE

 DESCRIPTOR--* DATA STRUCTURE OPERATORS
 DESCRIPTION-* FOCUS ON PROGRAMMER, RECORD AT A TIME, NAVIGATIONAL AID
    S

 DESCRIPTOR--* COMPOUND DATA STRUCTURE OPERATORS
 DESCRIPTION-* BUILT UP FROM PRIMITIVE OPERATORS
```

ID YYAUTHOR: 74SHNEIDERMAN + SCHEUERMANN
TITLE: "Structured Data Structures"

Purpose of article is to develop a top down, structured approach (analogous to structured programming) for data definition and manipulation. Many problems occur because languages allow user (programmer) to define any data structure and perform any operations on them. This approach defines a set of data structures (lists, trees, rings, stacks, queues, and dequeues) and the set of meaningful operations on them which the user can perform. Structures can be hierarchically nested (e.g., a list where each node is a queue and each queue element is a tree). Proposed DSDML (Data Structure Description and Manipulation Language) has four components: (1) definition -- of data structure; (2) search method -- access paths; (3) manipulation -- set of legal operations on each structure (primitives); and (4) extended manipulation -- frequently used sets of operations built from several primitives.

Article focuses on data structuring for storage and access. Very procedurally oriented aids for programming user in navigating through and operating on data structure. The non-programming end user probably doesn't need to know most of the details covered. However, in some cases this structuring could be useful to the end user, e.g., defining a FIFO or LIFO inventory. Also in some cases business policy defines an access path, e.g., ship the part from the nearest warehouse. Therefore, from a logical perspective the user may want some of these capabilities, but in using them he should not be implying the corresponding storage structure. Also a problem occurs if too many restrictions are placed on the legal operations for a structure. If the legal operations for stacks and queues only allow operations at the ends, how do you correct an erroneous entry in the middle?

```
ID YYAUTHOR-* 74SKINNER
TITLE-------* ≠A HEURISTIC APPROACH TO INDUCTIVE INFERENCE IN FACT RETR
     IEVAL SYSTEMS≠
SERIAL TITLE-------* COMMUNICATIONS OF ACM
SERIAL NUMBER------* 17:12
PUBLISHER----------* ACM
YEAR---------------* 1974
DATE---------------* 1974/12
PAGES--------------* 707-712

  AUTHOR NAME-* SKINNER
  FIRST NAME--* C. WILLIAM

  DESCRIPTOR--* INFERENTIAL RETRIEVAL
  DESCRIPTION-* FOR DERIVING MORE THAN STORED FACTS

  DESCRIPTOR--* QUERY TYPES

  DESCRIPTOR--* QUERY RESPONSES

  DESCRIPTOR--* RULES OF INFERENCE
  DESCRIPTION-* SIMILARITY STRUCTURE, LEVELS OF SIGNIFICANCE

  DESCRIPTOR--* VALIDATION CRITERIA
  DESCRIPTION-* CAN BE USED TO DETECT INCONSISTENCIES WHERE DEFINED SIM
      ILARITY STRUCTURES INDICATE OTHERWISE
```

ID YYAUTHOR: 72SKINNER

TITLE: "A Heuristic Approach to Inductive Inference in Fact Retrieval Systems"

Given a query of the form "what is attribute A for entity E," a "fact retrieval system" will give you the value if it is stored, otherwise it will simply tell you the query cannot be answered. An inferential system goes a step further and compares the entity with other similar entities and tries to infer, based on the degree of similarity, the value of the entity. The paper discusses the additions needed to convert a fact system into an inferential one. In addition to the usual database information the system also needs a similarity structure, defined by the user, which allows the system to cluster similar entities in terms of like attributes. User must specify the attributes on which to consider similarity and the levels of significance required before two entities will be considered similar enough on which to base an inference. (Article describes the actual clustering algorithm.) Entities are more similar if: (1) more of their attributes have the same value; and/or (2) those attributes on which they agree have a high variance among entities.

Four types of queries: (1) Is it true that a given entity has a given property (attribute/value pair)? (2) For a given entity, what attribute has a given value? (3) What entity has a given property? and (4) For a given entity, what is the value of a given attribute? Three types of responses: (1) the query is true or false; (2) the entity or attribute value requested; or (3) no response (the query cannot be answered with the available data). Inferential system will shift more responses from three into one and two. However, response types one and two should be qualified by their reliability or confidence, which is provided by the similarity structure (or because the actual value is in the database -- a fact retrieval).

Inference useful when missing data or inconsistent data, perhaps from several sources. User must be provided with a way to define the similarity structure, levels of significance, and to change them. Level of significance depends on the user. However, unclear whether similarity structure should be at schema or subschema level.

```
IC YYAUTHOR-* 74SNUGGS + POPEK + PETERSON
TITLE-------* #DATA BASE SYSTEM OBJECTIVES AS DESIGN CONSTRAINTS#
SERIAL TITLE-------* DATA BASE
SERIAL NUMBER------* 6:3
PUBLISHER----------* ACM
YEAR---------------* 1974
DATE---------------* 1974 WINTER
PAGES--------------* 11-20

   AUTHOR NAME-* SNUGGS
   FIRST NAME--* MARY

   AUTHOR NAME-* POPEK
   FIRST NAME--* GERALD

   AUTHOR NAME-* PETERSON
   FIRST NAME--* RONALD

   DESCRIPTOR--* DATABASE SYSTEM DESIGN OBJECTIVES

   DESCRIPTOR--* OBJECTIVES
   DESCRIPTION-* IN DESIGN OF DATABASE SYSTEM

   DESCRIPTOR--* PERFORMANCE MONITORING

   DESCRIPTOR--* PERFORMANCE CRITERIA
   DESCRIPTION-* TIME, SPACE, VOLATILITY, RESPONSE TIMES

   DESCRIPTOR--* DATA INTEGRITY
   DESCRIPTION-* BACKUP AND RECOVERY, CONCURRENCY CONTROL, ACCESS CONTRO
        L, ENCRYPTION

   DESCRIPTOR--* EVOLVABILITY

   DESCRIPTOR--* DESIGN TRADEOFFS
   DESCRIPTION-* AMONG DATABASE SYSTEM OBJECTIVES, USER MUST GIVE SUFFIC
        IENT INFORMATION TO ENABLE TRADEOFFS TO BE MADE
```

ID YYAUTHOR: 74SNUGGS + POPEK + PETERSON

TITLE: "Data Base System Objectives as Design Constraints"

Seven basic data base objectives. One of key tasks for data base design is to determine their relative importance.

1. Data independence -- to allow the physical restructuring of data while not affecting the application programs. The logical form of the data and access paths through it should be unaffected by physical changes.
2. Data relatability -- system should automatically maintain data base consistency by propogating the appropriate changes through the data. Applies to derived items and inter-item or group validation criteria. Responsibility for this function split between application program and system. Most of the load should be on the system once the user has defined the relationships to be maintained.
3. Compatibility -- ease of conversion between different hardware and software environments when a system is replaced or upgraded. Also a problem with linkage between several "independent" DMS being used at the same facility.
4. Structural adaptability -- the ability of the system to restructure itself in response to changes in behavior. To know when to change requires performance monitoring and user specified criteria. To be adaptable and able to change requires both data independence and flexibility and generality in the system.
5. Data integrity -- four causes of errors in the data base which must be monitored: (1) input errors; (2) hardware failures; (3) software bugs; and (4) shared access (concurrent updates, deletes by one user when others still need data, and deadlock). Redundancy required for both preventing errors (integrity) and correcting them (recovery).
6. Data recovery -- two basic functions: (1) to prevent error propogation through the data base (requires early detection either automatically or with highly visible conditions for manual intervention); and (2) to repair the errors that have occurred.

ID YYAUTHOR: 74SNUGGS + POPEK + PETERSON
TITLE: "Data Base System Objectives as Design Constraints"

7. Data security -- proposes isolating the security processes in a security kernel. With greater use of public communications, security measures such as encryption becoming more important. Access control must allow data dependent criteria (instance level or value related criteria as opposed to schema or data item level criteria).

System performance must be monitored for: (1) resource allocation (time, space, and volatility); (2) response times (for queries and updates); and (3) other user specified expectations (such as ease of use, currency, accuracy). Finally, trade-offs must be made between the basic objectives and performance: (1) data independence vs. response time; (2) redundancy (for integrity and recovery) vs. storage space; (3) security checks vs. response time; (4) generality and adaptability vs. time and space; and (5) volatility and reorganization vs. efficiency in both the long and short term.

One of the basic design tasks is to determine the relative importance of the seven objectives and what price should be paid in terms of performance. These are user decisions and a way must be found to allow him to make these specifications to the design process in terms that are meaningful to him.

```
ID YYAUTHOR-# 74TAGGART
TITLE-------# #DEVELOPING AN ORGANIZATION#S INFORMATION INVENTORY#
SERIAL TITLE-------# MANAGEMENT INFORMATICS
SERIAL NUMBER------# 3:6
PUBLISHER----------# IFIP ADP GROUP (IAG)
PUBLISHER LOCATION-# AMSTERDAM
YEAR---------------# 1974
DATE---------------# 1974/12
PAGES--------------# 283-292

   AUTHOR NAME-# TAGGART
   FIRST NAME--# WILLIAM JR.

   DESCRIPTOR--# DATA ITEM DEFINITION
   DESCRIPTION-# ATTRIBUTE, TIME, ENTITY DESCRIBED

   DESCRIPTOR--# TIME
   DESCRIPTION-# ANOTHER DIMENSION ON THE STORED DATABASE

   DESCRIPTOR--# DATA DICTIONARY
   DESCRIPTION-# DEFINITION OF INFORMATION ELEMENTS, USE OF DATA DICTION
        ARY

   DESCRIPTOR--# REPORT DEFINITION
   DESCRIPTION-# USING CONCEPT OF INFORMATION ELEMENT

   DESCRIPTOR--# INFORMATION ALGEBRA

   DESCRIPTOR--# STATUS DATA
   DESCRIPTION-# LEVEL OR STATE AT A POINT IN TIME

   DESCRIPTOR--# FLOW DATA
   DESCRIPTION-# CHANGE OVER TIME PERIOD, RATE OF CHANGE, EVENT INFORMAT
        ION
```

ID YYAUTHOR: 74TAGGART
TITLE: "Developing an Organization's Information Inventory"

Main proposition that a data item is inadequate to identify information. Develops the concept of information element, which includes: (1) the attribute of interest; (2) time; and (3) the entity. Concept evolved from four areas: (1) information algebra -- more than a single property should be present to identify information; (2) Chapin -- any attribute can be the focal point of interest and the other attributes serve as identifiers; (3) Langefors -- four essential components -- the attribute of interest, its value, the time for which it applies, and the entity; and (4) McDonough -- more than one set of attributes may be needed to completely identify the entity of interest.

Fundamental difference in state and event attributes based on how they consider time. State applies to a point in time. Event refers to the change in a state over a period of time.

Information element characteristics include: (1) entity component size (McDonough); (2) sequenced entity terms -- used to order a set of entities; (3) non-sequenced entity terms -- used only to identify not order entities; and (4) detail versus summary information elements -- indicates over which entity sets the summary occurs (if five attributes are required to completely identify an entity, then the presence of only four implies a summary over the fifth attribute).

Systems such as ADS and TAG have problems because they don't allow data items to be grouped into user relevant information elements and don't adequately handle time.

Proposes the use of an information inventory in terms of information elements, in addition to the usual data dictionary. Reports, regardless of whether they are for time points or periods, are defined as collections of information elements. Suggests three uses for information inventory: (1) identify information duplication in reports; (2) determine if new needs can be met by information already being produced; and (3) help the user clarify new information requirements (tries to help the user who has a general idea of what he wants, but doesn't know specifically what to ask for).

ID YYAUTHOR: 74TAGGART
TITLE: "Developing an Organization's Information Inventory"

Two questions raised in considering article: (1) when defining his problem should the user define his information requirements in terms of information elements instead of or in addition to the usual data items and groups; and (2) are information elements an appropriate way to define reports?

```
ID YYAUTHOR-# 75BURROUGHS
TITLE-------# DMSII DATA AND STRUCTURE DEFINITION LANGUAGE(DASDL)REFERE
    NCE MANUAL
PUBLISHER----------# BURROUGHS
YEAR---------------# 1975

  AUTHOR NAME-# BURROUGHS

  DESCRIPTOR--# DATABASE MANAGEMENT SYSTEM
  DESCRIPTION-# DBTG IMPLEMENTATION--DIFFERENCES REPRESENTATION OF INTE
       RFILE RELATIONSHIPS AND COMBINED LOGICAL AND PHYSICAL DEFINITIO
       N

  DESCRIPTOR--# ACCESS PATH
  DESCRIPTION-# DIRECT, CALC, VIA

  DESCRIPTOR--# DATA ITEM DEFINITION
  DESCRIPTION-# NAME, TYPE, SIZE, INITIAL AND NULL VALUES, NUMBER OF OC
       CURRENCES IF REPEATING, AND MANDATORY OR OPTIONAL

  DESCRIPTOR--# DATA DEFINITION
  DESCRIPTION-# LOGICAL AND PHYSICAL STRUCTURE AND VALIDATION CRITERIA
       FOR INCLUSION OF RECORD IN THE FILE

  DESCRIPTOR--# BACKUP AND RECOVERY
```

ID YYAUTHOR: 75BURROUGHS

TITLE: "DMSII Data and Structure Definition Language (DASDL) Reference Manual"

DMSII is an implementation of a variation of the DBTG specifications. DASDL provides the Database Administrator with the facilities to completely specify the logical and physical structure of the database.

Files or "data sets" can be organized as DIRECT, RANDOM (which includes DBTG's CALC and VIA), RESTART (special system database used for recovery), and INORDERED. For each file there can be any number of "sets" or pointer arrays or indexes providing access paths to the file. Records within a file are defined by name, type, size, and VERIFY (specifying a condition to be tested to insure a record's membership in the file). Data item definition includes name, type (alphabet, boolean, and signed or unsigned integer or real), size, an initial and a null value, the number of times the item occurs in the record if it is repeating, and whether or not the item is required. Items can be grouped into groups by specifying the item names, the number of occurrences and whether or not the item is required.

For RANDOM files DMSII provides five types of linkages: (1) counted -- the number of links to a record is maintained and the record can only be deleted when the count has been reduced to zero; (2) self-correcting -- when the record has been obtained (through its disk address stored in the "owner" record) the key is checked. If there is not a match, the appropriate index or set is checked to find the proper record and the pointer in the parent is corrected; (3) symbolic link -- the linkage is through a set of index with the key value rather than directly with the disk address; (4) unprotected linkage -- the disk address pointer in the "owner" record is assumed correct and no checks are made; and (5) verified link -- when the record is obtained (through a disk address) a check is made to verify that there is agreement on the particular key item value (the data item being checked does not have to be a key item).

ID YYAUTHOR: 75BURROUGHS

TITLE: "DMSII Data and Structure Definition Language (DASDL) Reference Manual"

Access can be on any key item, which can be specified as ascending or descending, whether or not duplicates are allowed, and whether inserts should be make first or last or in order in a data set. Access can be indexed sequential, indexed random, through ordered or unordered lists, or with a bit vector. Since most of the links are through disk addresses, there are several optional checks to insure that the records obtained are the ones sought.

When defining the database it can be specified as AUDIT or recoverable. Also, synchronization and control points can be specified for roll back and recovery.

```
ID YYAUTHOR-* 75COMPUTER ASSOCIATES
TITLE-------* EARL GENERAL INFORMATION MANUAL
PUBLISHER----------* COMPUTER ASSOCIATES
PUBLISHER LOCATION-* NEW YORK, NY.
YEAR---------------* 1975
  AUTHOR NAME-* COMPUTER ASSOCIATES
  FIRST NAME--* INC.

  DESCRIPTOR--* DATABASE MANAGEMENT SYSTEM

  DESCRIPTOR--* FLAT FILE

  DESCRIPTOR--* REPORT DEFINITION
  DESCRIPTION-* DEFAULT LABELING AND EDITING. VALUE DEPENDENT OUTPUT CO
       NTROL
```

ID YYAUTHOR: 75COMPUTER ASSOCIATES

TITLE: EARL - General Information Manual

Earl is a report writer using flat files with extensive defaults which can be overridden.

It has value dependent control over item printing. A tag is set to a value by an if statement. The value of this tag is associated with an item in the output list. Those items with the same value as the tag are printed. Thus, the same item can be printed in different columns. Several values of the tag can be concatenated implying a logical OR (print item if tag has any of the values).

Defaults are item names for column titles, input format for output format.

```
ID YYAUTHOR-* 75ESWARAN + CHAMBERLIN
TITLE--------* #FUNCTIONAL SPECIFICATIONS OF A SUBSYSTEM FOR DATA BASE I
     NTEGRITY#
SERIAL TITLE--------* PROCEEDINGS OF INTERNATIONAL CONFERENCE ON VERY LA
     RGE DATA BASES
EDITOR---------------* KERR, DOUGLAS S.
YEAR-----------------* 1975
DATE-----------------* 1975/09
PAGES----------------* 48-68

  AUTHOR NAME-* ESWARAN
  FIRST NAME--* KAPALI P.

  AUTHOR NAME-* CHAMBERLIN
  FIRST NAME--* DONALD D.

  DESCRIPTOR--* DATA DEFINITION
  DESCRIPTION-* PROPOSES DYNAMIC MODIFICATION OF VALIDATION CRITERIA

  DESCRIPTOR--* DATA INTEGRITY
  DESCRIPTION-* FUNCTIONS INCLUDE ACCESS CONTROL, CONCURRENT UPDATE CON
       TROL, BACKUP AND RECOVERY, AND VALIDATION

  DESCRIPTOR--* ACCESS CONTROL
  DESCRIPTION-* BY FUNCTION, ESPECIALLY FOR MODIFYING VALIDATION CRITER
       IA

  DESCRIPTOR--* VALIDATION RESPONSE
  DESCRIPTION-* WHEN VALIDATION CRITERIA VIOLATED--HARD(REJECT TRANSACT
       ION)OR SOFT (ISSUE MESSAGE TO USER)

  DESCRIPTOR--* DATA ITEM OPERATORS
  DESCRIPTION-* DEFINE ITEM TYPES WHICH CAN BE VALID OPERANDS

  DESCRIPTOR--* SEMANTIC CONSTRAINTS
  DESCRIPTION-* AS VALIDATION CRITERIA

  DESCRIPTOR--* VALIDATION CRITERIA
  DESCRIPTION-* TO WHAT THEY APPLY, HOW THEY APPLY, WHEN THEY APPLY, AN
       D WHEN TO INVOKE THEM

  DESCRIPTOR--* CONDITIONS ON A PROCESS
  DESCRIPTION-* FOUR CLASSIFICATIONS-(1)ITEM DOMAIN VS INTER-ITEM CONSI
       STENCY, (2)STATE TRANSITION VS TRANSACTION(3)IMMEDIATE(ALWAYS H
       OLDS) VS DELAYED(SPECIFY WHEN TO TEST),AND(4)INVOKED ON DEMAND
```

ID YYAUTHOR: 75ESWARAN + CHAMBERLIN
TITLE: "Functional Specifications of a Subsystem for Data Base Integrity"

There are four subsystems to prevent errors in a database: (1) security -- prevents unauthorized access; (2) consistency -- controls concurrent updates; (3) reliability -- relates to hardware/software failures; and (4) integrity -- prevents errors due to lack of knowledge or careless.

Integrity is based on a set of premise-action statements which apply to the database. They are a subset of those premises which apply in the real world, but are limited by lack of knowledge and language capacity for expressing them.

An integrity subsystem is proposed which allows for the definition of premises (or assertions of correctness) and for the actions when the integrity has been violated. Definition of premises are separate from data definition and are in a language similar to the query language. The data definition is static, but premises (assertions) are dynamic and user can change them (ignores effects of these changes on database integrity). Needs new type of access authorization to modify premises. *Proposes a premise catalog* (like with data definition) to answer what premises are in effect or what premises apply to a particular data item. May be desirable to allow premises to attach to either schema or subschema (let user be more restrictive).

Classification of integrity checks:

- to what they apply -- entities versus sets of entities (tuple vs set)
- how they apply -- state or transitional to values - with or without regard to past value
- when they apply (or are enforced)

  immediate -- always enforced
  delayed -- at end of a transaction
  selectively -- at user discretion
- when they are invoked -- triggered by an update operation

The actions when a premise fails may be hard (reject the transaction) or soft (make the change but issue a warning).

ID YYAUTHOR: 75ESWARAN + CHAMBERLIN

TITLE: "Functional Specifications of a Subsystem for Data Base Integrity"

Actions on a failure of an assertion:

- the user must be given enough information to "correct" the problem
- immediate assertions

  return assertion violated and tuples violating assertion
  actions skip transaction, change request, or roll back
- delayed assertions

  return error code (can't determine which tuples caused error)
  actions must back out
- soft assertions -- probable errors or errors which are allowed to exist but with proper notification to users
- null values

  treated as information void
  shouldn't cause assertion to succeed if it would have failed or fail if it would have succeeded

Set of compatible items may be defined for which certain operations are allowed, even though prior conversion may be necessary. Examples, arithmetic operations allowed between integers and real after conversion, but not between arithmetic items and charcter strings. Extreme example would be dimensional analysis of arithmetic operations.

```
ID YYAUTHOR-* 75GOODENOUGH
TITLE-------* *EXCEPTION HONDLING: ISSUES AND A PROPOSED NOTATION*
SERIAL TITLE-------* COMMUNICATIONS OF THE ACM
SERIAL NUMBER------* 18:12
PUBLISHER----------* ACM
YEAR---------------* 1975
DATE---------------* 1975/12
PAGES--------------* 683-696

  AUTHOR NAME-* GOODENOUGH
  FIRST NAME--* JOHN B.

  DESCRIPTOR--* EXCEPTION PROCESSING
  DESCRIPTION-* SYSTEM DEFAULTS OR USER SPECIFIED

  DESCRIPTOR--* EXCEPTION CONDITIONS
  DESCRIPTION-* RANGE FAILURE, DOMAIN FAILURE, RESULTS, OR MONITORING

  DESCRIPTOR--* EXCEPTION CONDITION RESPONSES
  DESCRIPTION-* ABORT JOB AND ROLL-BACK, TRY AGAIN, NOTIFY INVOKER OF P
         ARTIAL RESULTS
```

ID YYAUTHOR: 75GOODENOUGH

TITLE: "Exception Handling: Issues and a proposed Notation"

This paper presents a very flexible method for handling exceptions. Exceptions are conditions that are brought (raising an exception) to an invoker's attention. The response to an exception i[illegible] handling the exception. The notion of an exception is very important in information systems. There are many reasons why a normal path of processing can be interrupted. The user needs to specify conditions for and responses to exception processing situations.

Exceptions can be raised because of:

- range failure -- the operation cannot satisfy its output assertion
- domain failure -- an input assertion cannot be satisfied by a process
- result classification -- provide additional information about the output
- monitoring -- notify invoker that a condition has occurred

In the above exception conditions, the invoker should be allowed to:

1. abort the operation and undo its effects
2. try the operation again
3. terminate operation with partial results

An operation (subroutine, operator, etc.) can cause an exception. It can be raised by the user or the system. Every operation should declare what exceptions it can raise. Handlers for exceptions can be declared with the operation raising the exception or with an other operation within the scope of the operation raising the exception.

Three ways to classify exceptions are described with appropriate responses. These distinctions allow for some compile time checks.

ESCAPE -- requires termination of operation

NOTIFY -- forbids termination of operation

SIGNAL -- terminate or continue operation

Each exception handler is declared to be one of the above types. The same terms are used to invoke the handler.

ID YYAUTHOR: 75GOODENOUGH
TITLE: "Exception Handling: Issues and a Proposed Notation"

When leaving an exception handler, the last statement must state termination or continuation of the operation causing the exception. This must be consistent with the type of the handler.

EXIT -- Terminate the operation (ESCAPE & SIGNAL)
A valued exit will send back a value for the operation being terminated

RESUME -- Resume execution after the operation causing the exception

A handler has the ability to handle other exceptions within itself.

ENDED -- An exception raised upon normal completion of an operation
An ESCAPE type exception

CLEANUP -- An exception raised when an operation is not resumed

Exceptions can have default handlers which are systems or user defined. The user should have the ability to override these default handlers. By declaring a handler to be OPTIONS a user may substitute his own handler for it.

A way should be provided to get back to the default handler. RESUME (DEFAULT) will invoke the default handler.

An other condition called DEFAULT can be raised in a handler.

A PASS will pass an exception condition to an invoker and let him handle it. An exception is propogated upward.

Notation is cumbersome and hard to understand.

```
IC YYAUTHOR-* 75HAMMER + MCLEOD
TITLE-------* ≠SEMANTIC INTEGRITY IN A RELATIONAL DATA BASE SYSTEM≠
SERIAL TITLE-------* PROCEEDINGS INTERNATIONAL CONFERENCE ON VERY LARGE
      DATA BASES
EDITOR--------------* KERR, DOUGLAS S.
YEAR----------------* 1975
DATE----------------* 1975/09
PAGES---------------* 25-47

  AUTHOR NAME-* HAMMER
  FIRST NAME--* MICHAEL J.

  AUTHOR NAME-* MCLEOD
  FIRST NAME--* DENNIS J.

  DESCRIPTOR--* VALIDATION CRITERIA
  DESCRIPTION-* STATE-VALID DOMAINS AND RELATIONS CHECKED. TRANSITION-IL
        EGAL OPERATORS. LANGUAGE TO STATE AND FUNCTION TO STORE CRITERI
        A

  DESCRIPTOR--* VALIDATION PROCESS
  DESCRIPTION-* TO TEST STORED CRITERIA AGAINST DATA AND INITIATE EXCEP
        TION RESPONSE

  DESCRIPTOR--* VALIDATION RESPONSE

  DESCRIPTOR--* SEMANTIC INTEGRITY
  DESCRIPTION-* DISCUSSES TWO TYPES-STATE AND TRANSITION
```

ID YYAUTHOR: 75HAMMER + MCLEOD
TITLE: "Semantic Integrity in a Relational Data Base System"

Discusses two types of semantic integrity (state and transition), requirements for the subsystem, and the components of domain and relational constraints or premises. Although written from a relational perspective, it would apply to any data model.

State integrity insures valid domain and relationships for items and sets. Transition integrity insures that only valid operations done on values and sets. Transition integrity (related to abstract data types) not really discussed.

Validity definition (or integrity subsystem) requires: (1) high level, non-procedural language; (2) processor for validity (of constraint) expression language; (3) validity enforcer -- to decide when to test premise and to actually test it; (4) action processor -- to specify exception process; and (5) constraint compatibility tester. Language must allow definition of premise and action. Premise must specify both validation criteria and when it applies or should be tested. Specification of when allows database to pass through temporarily invalid states. Premises may apply to either item domains or set membership and relationships. Authors state separate languages required for stating domain and relation premises, but the reason is not apparent.

```
IC YYAUTHOR-* 75TSICHRITZIS
TITLE-------* LSL: A LINK AND SELECTOR LANGUAGE
MCNOGRAPH TITLE----* TECHNICAL REPORT
MCNOGRAPH NUMBER---* CSRG-61
PLBLISHER----------* COMPUTER SCIENCE DEPARTMENT-UNIVERSITY OF TORONTO
PLBLISHER LOCATION-* TORONTO, CANADA
YEAR---------------* 1975

   AUTHOR NAME-* TSICHRITZIS
   FIRST NAME--* D.

   DESCRIPTOR--* HIGH LEVEL MULTIFILE LANGUAGE

   DESCRIPTOR--* DATA DEFINITION
   DESCRIPTICN-* SCHEMA, SUBSCHEMA, VALIDITY CRITERIA, AND ACCESS PATHS

   DESCRIPTOR--* USERSCHEMA
   DESCRIPTICN-* USER VIEW OF DATA

   DESCRIPTOR--* ACCESS PATH
   DESCRIPTICN-* DYNAMICALLY CREATED, MAINTAINED, AND DESTROYED

   DESCRIPTOR--* DATABASE OPERATORS

   DESCRIPTOR--* VALIDATION
```

ID YYAUTHOR: 75TSICHRITZIS
TITLE: "LSL: A Link and Selector Language"

LSL is a DBMS offering multiple views of data. By using a small set of primitives a user can define a view and the constraints which apply to the view.

The basic structures are record type, selector, link, and expression. Each has a generation statement defining how to create it and an intention statement defining constraints on the structure.

A record type is a set of entities composed of items some of which are candidate keys. Generation defines how it is populated. Intention defines constraints (value check, item relationships) or validation within a record.

A link relates two record types -- defines an access path. Generation defines how link is created. Intention defines mapping (1 to 1, N to M).

A selector defines a subset of a record type by a boolean selection expression. Generation defines those entities in the subset and intention defines the properties of the entities.

An expression used for the creation of access paths to a subset of the DB (between many record types) using links and selectors.

Intentions and generations for links and selectors are under user (DBA) control. He can create them when needed with or without systems maintenance (dynamically when needed or under systems control) or create and lock out records involved. Links are created bidirectionally but are deleted in one direction.

The user navigates through the structure using a calculus-like language which operates upon groups of records rather than one record at a time.

A relation (hit file) is the result of a query. It is created by using links and selectors and keeping items.

ID YYAUTHOR: 75TSICHRITZIS
TITLE: "LSL: A Link and Selector Language"

LSL Makes a definite distinction between defining a structure and creating it.

Missing concepts: if an intention is not met nothing is provided to handle the exceptions. No definition of inter-record constraints.

```
ID YYAUTHOR-* 75UNIVAC
TITLE-------* GLP USERS REFERENCE MANUAL
PUBLISHER----------* UNIVAC INC.
PUBLISHER LOCATION-* ROSEVILLE, MINNESOTA
YEAR----------------* 1975

   AUTHOR NAME-* UNIVAC
   FIRST NAME--* INC.

   DESCRIPTOR--* DATABASE MANAGEMENT SYSTEM

   DESCRIPTOR--* HIGH LEVEL MULTIFILE LANGUAGE
```

ID YYAUTHOR: 75UNIVAC
TITLE: QLP - Users Reference Manual

QLP is an update/query language that must interface to DMS-1100 and a database.

QLP has operations to protect the database against inadvertant changes. A HOLD freezes the database and makes the current state as a hold point. All changes to the database are temporary. RELEASE will make all changes permanent while ROLLBACK rolls the database back to the state of the last HOLD.

The selection clause has an in or out of range check.

```
IC YYAUTHOR-* 75WEBEH
TITLE-------* #AUDIT CAPABILITIES UF SOME DATABASE MANAGEMENT SYSTEMS#
SERIAL TITLE-------* CUNFERENCE ON COMPUTER, AUDIT, CONTRCL AND SECURIT
     Y
SERIAL NUMBER------* FIFTH ANNUAL
YEAR---------------* 1975
DATE---------------* 1975/04

   AUTHOR NAME-* WEBER
   FIRST NAME--* RCN

   DESCRIPTOR--* SURVEY OF DMS
   DESCRIPTION-* CCMPARES 20 SYSTEMS

   DESCRIPTOR--* AUDITOR
   DESCRIPTICN-* RCLE IN DMS. USE OR NON-USE OF DMS FOR AUDIT FUNCTION

   DESCRIPTOR--* FEATURE LIST
   DESCRIPTION-* FCR DMS
```

ID YYAUTHOR: 75WEBER

TITLE: "Audit Capabilities of Some Database Management Systems"

Basic role of auditor (both internal and external) is to validate the data and its interpretation. Many audit packages for nondatabase systems. Database creates problems auditors have not yet resolved. If auditing requests are made through the DMS, then it is not a totally independent analysis. If auditing requests do not go through DMS, then all access to database are not being made and controlled through DMS, creating a potential threat to database integrity. Three possible approaches suggested and evaluated: (1) modify existing audit software; (2) establish an interface between current packages and the DMSs; or (3) design audit capability into the DMSs. Depending on the methods a DMS uses to recover the database, there is a similarity in the data required for auditing and recovery.

Paper compares a number of systems (ASI-ST, IDMS, Data Analyzer, DYL-260, EASYTRIEVE, GIM-II, GIS, IMS/90, INQUIRE, INSYTE, MANAGE, MARK IV, MARS VI, Model 204, OLIVER, QUERY, UPDATE, RAMIS, SYSTEM 2000, and TOTAL/SOCRATES) on several capabilities. The capabilities include data definition and redefinition, data structure, ability to handle synonyms, validation rules, retrieval commands for selection, sampling, extracting, file and entry level derivation, sorting, and report definition and output editing.

```
ID YYAUTHOR-* 76ASTRAHAN
TITLE-------* #SYSTEM R: RELATIONAL APPROACH TO DATABASE MANAGEMENT#
SERIAL TITLE-------* TODS
SERIAL NUMBER------* 1:2
PUBLISHER----------* ACM
YEAR---------------* 1976
DATE---------------* 1976/06
PAGES--------------* 97-137

  AUTHOR NAME-* ASTRAHAN
  FIRST NAME--* M. M., ET. AL.

  DESCRIPTOR--* DATABASE MANAGEMENT SYSTEM
  DESCRIPTION-* IBM RESEARCH SYSTEM

  DESCRIPTOR--* ACCESS CONTROL
  DESCRIPTION-* USER GRANTS AUTHORIZATION FOR READ, CHANGE, INSERT, DEL
        ETE, CREATE LINKS OR VIEWS

  DESCRIPTOR--* ACCESS PATH
  DESCRIPTION-* DEFINITION OF INDEXES(IMAGES)AND INTER-ENTITY RELATIONS
        HIPS(LINKS)

  DESCRIPTOR--* USERSCHEMA

  DESCRIPTOR--* TRIGGER

  DESCRIPTOR--* TRANSACTION DEFINITION
  DESCRIPTION-* A SET OF SEQUEL COMMANDS

  DESCRIPTOR--* HIGH LEVEL MULTIFILE LANGUAGE
  DESCRIPTION-* SEQUEL FOR QUERY AND UPDATE

  DESCRIPTOR--* DATA INTEGRITY
  DESCRIPTION-* MAINTAINED BY ASSERTIONS WHICH DESCRIBE PERMISSIBLE STA
        TES OF THE DATABASE OR PERMISSIBLE TRANSITIONS OF DATABASE

  DESCRIPTOR--* SEMANTIC CONSTRAINTS
  DESCRIPTION-* CALLED ASSERTIONS

  DESCRIPTOR--* VALIDATION CRITERIA
  DESCRIPTION-* ACCEPTABLE STATES OR TRANSITIONS. ENFORCED AT END OF TR
        ANSACTION OR WHEN USERINVOKES THEM. ROLL BACK WHEN FAILURE

  DESCRIPTOR--* CONCURRENT CONTROL
```

ID YYAUTHOR: 76ASTRAHAN

TITLE: "System R: Relational Approach to Database Management"

System R is a multi-user relational DBMS developed by IBM as a research vehicle.

The SEQUEL language is the interface to the database for both the programmer and non-programmer. The programmer calls SEQUEL with his SEQUEL command as a string. Local variables can be included in this text and can be bound to their local addresses (within the program).

Access paths (for performance) between relations can be defined as images or links. An image is an index which can be declared to only have unique values. A link is a physical binary link between tuples (one to many) based upon a value in certain fields. Both structures can have a clustering property where tuples with common values are physically adjacent.

A subschema capability called the VIEW is provided. It is a definition of a SEQUEL statement on other relations. This VIEW can be used in a query or update (only if there is a 1 to 1 relationship between tuples in the view and tuples in the underlying relation).

A transaction is a set of SEQUEL commands that are processed as one command. The user may specify save points and may back out (undoing all operations) to any save point.

The creator of an object (relation) has the ability to control the object. He can grant authorization for its use to other users who in turn can grant their authorizations. These authorities include read, insert, delete, update, drop (delete a relation), expand (add domains), create images and links, control (define views and triggers) and grant (give above permissions to other users).

Integrity is maintained by assertions which describe permissible states of the database or permissible transitions of the database. Assertions are enforced at the end of a transaction or the user can explicitly invoke them. If an assertion is not met the system backs out the transaction. The user defines

ID YYAUTHOR: 76ASTRAHAN

TITLE: "System R: Relational Approach to Database Management"

that an assertion be invoked upon an insertion, deletion or update of a relation or domain.

A trigger is similar to an assertion. Instead of containing a predicate it contains a sequence of SEQUEL statements. It is also invoked under the same conditions plus the read of a tuple or domain.

When controlling for concurrency System R provides three levels of transaction consistency:

1. least isolation from other users
   read dirty data and read different values for
   same item during transaction
2. access clean data (data not being updated)
   subsequent access may give different results
3. access clean data and reads are reproducible --
   logically a single user system

When deadlock is detected the youngest transaction with the shortest duration locks is backed out to the last save point.

```
ID YYAUTHOR-* 76DIJKSTRA
TITLE-------* A DISCIPLINE OF PROGRAMMING
PUBLISHER----------* PRENTICE-HALL, INC
PUBLISHER LOCATION-* ENGLEWOOD CLIFFS, NEW JERSEY
YEAR---------------* 1976

  AUTHOR NAME-* DIJKSTRA
  FIRST NAME--* EDSGER W.

  DESCRIPTOR--* PREMISE ACTION

  DESCRIPTOR--* CONDITIONS ON A PROCESS
```

ID YYAUTHOR: 76DIJKSTRA
TITLE: A Discipline of Programming

Dijkstra presents two control structures (guarded statements) which are nondeterministic.

IF{<guard> → <statement list} ... FI

A guard is a boolean expression. A guard is picked at random and evaluated until a true guard is found. Then the statements with that guard are performed.

DO{<guard> → <statement list>} ... OD

Similar to the IF except the DO is repeatedly executed until all guards are false.

```
ID YYAUTHOR-* 76GRIFFITHS + WADE
TITLE-------* ≠AN AUTHORIZATION MECHANISM FOR A RELATIONAL DATABASE SYS
      TEM≠
SERIAL TITLE-------* TODS
SERIAL NUMBER------* 1:3
PUBLISHER----------* ACM
YEAR---------------* 1976
DATE---------------* 1976/09
PAGES--------------* 242-255

  AUTHOR NAME-* GRIFFITHS
  FIRST NAME--* PATRICIA P.

  AUTHOR NAME-* WADE
  FIRST NAME--* BRADFORD W.

  DESCRIPTOR--* USERSCHEMA
  DESCRIPTION-* DEFINED AS VIEWS TO A RELATIONAL DATABASE. BASIS FOR AC
        CESS CONTROL

  DESCRIPTOR--* ACCESS CONTROL
  DESCRIPTION-* GRANTING AND REVOKING ACCESS PRIVILEGES. FORMING NETWOR
        K OF PRIVILEGES.

  DESCRIPTOR--* ACCESS PRIVILEGES
  DESCRIPTION-* FOR READ, INSERT, DELETE, UPDATE, OR DROP(ALSO IN TERMS
         OF ALL OR ALL BUT)

  DESCRIPTOR--* TIME STAMPING
  DESCRIPTION-* FOR GRANTING AND REVOKING PRIVILEGES
```

ID YYAUTHOR: 76GRIFFITHS + WADE

TITLE: "An Authorization Mechanism for a Relational Database System"

This paper addresses the problem of granting and revoking security in a multi-user environment. Authorization is discussed in terms of base relations (tables) and virtual relations (views).

The creator of a table can grant permission to use this table to others. The following operations are grantable:

read -- use table in a query and define views upon it
insert -- add new rows to a table
delete -- remove rows from a table
update -- modify specific columns of a table
drop -- delete a table

A creator can grant any of them or ALL or ALL BUT certain ones to any user, or can make them all PUBLIC. Also each user can be granted permission to grant their authorities to others. Each grantee can have grantable and nongrantable privileges from one or several sources on a single table. This leads to a network of granted authorities.

The concept behind revoking authority is to go to a state as if the grantee never received the privileges. This means revoking the privileges of the grantee and to all others granted privileges from the grantee.

A grantee can receive the same privileges from several sources or can receive the same privileges from one of his grantees. Two methods are presented to properly revoke privileges. A grant of privileges can be labeled and each subgrant of these privileges also has this label. By following the label through the granting process the proper privileges can be revoked. Each grant is time stamped. In the case where a user has received the same privileges from two different sources and granted them, the time stamp is used to revoke privileges properly.

ID YYAUTHOR: 76GRIFFITHS + WADE
TITLE "An Authorization Mechanism for a Relational Database System:

Views present different problems. Certain grantable privileges cannot be performed on a view. Authorizations for a views creator are developed by intersecting the authorizations of all underlying views and tables. The creator can only grant those authorizations he holds for all underlying tables and views.

If a view is dropped, all other views built (wholly or partially) upon the dropped view must be removed along with all privileges.

The paper does not address the following problems:

1. SYSTEM R has no provisions for a DBA. Therefore, no one controls the grantor with respect to what he may grant and to whom. Only a grantor can revoke or grant privileges.
2. A grantor cannot revoke all or specific privileges from all or selected users.

```
ID YYAUTHOR-* 76KAHN
TITLE-------* #A METHOD FOR DESCRIBING INFORMATION REQUIRED BY THE DATA
      BASE DESIGN PROCESS#
SERIAL TITLE-------* PROCEEDINGS SIGMOD CONFERENCE
EDITOR-------------* ROTHNIE, JAMES B.
PUBLISHER----------* ACM
YEAR---------------* 1976
DATE---------------* 1976/06
PAGES--------------* 55-64

  AUTHOR NAME-* KAHN
  FIRST NAME--* BEVERLY

  DESCRIPTOR--* DATABASE DESIGN
  DESCRIPTION-* INFORMATION REQUIREMENTS

  DESCRIPTOR--* DESIGN DATABASE
  DESCRIPTION-* CONTENT

  DESCRIPTOR--* INFORMATION REQUIREMENTS

  DESCRIPTOR--* DATA ITEM DEFINITION
  DESCRIPTION-* VALUE SET DISTRIBUTION

  DESCRIPTOR--* PROCESS DEFINITION
  DESCRIPTION-* RECORD-LEVEL VERBS, BEHAVIORAL CHARACTERISTICS

  DESCRIPTOR--* BEHAVIORAL CHARACTERISTICS
  DESCRIPTION-* DATA VOLUME, ACCESS AND PERFORMANCE STATISTICS, INTEGRI
         TY CRITERIA, ACCESS CONTROL, PROCESS PRECEDENCE ORDERING, USE F
         REQUENCY, TIMING
```

ID YYAUTHOR: 76KAHN

TITLE: "A Method for Describing Information Required by the Database Design Process"

The article identifies three steps in database design:

1. definition of information requirements for the design process
2. consideration of the information requirements for the perspectives of:
    a. information structure
    b. usage
3. integration of the two perspectives

The focus is on the information structure and usage perspectives. From the information perspective there are entities, attributes and relations. For each entity type the information needed includes volume, access, performance statistics, integrity criteria, and access controls. For attributes or data items, the information includes length, value set, average and maximum number of occurrences, probability of null values and access controls. For relations (which may be one to one, one to many, or many to many) there is a need to know the cardinality, integrity criteria, access control, and mapping functions.

From the usage perspective there is a need to: (1) develop a top down identification of process structure and process data interactions; (2) specify how processes use data in terms of a limited set of primitives; and (3) provide detailed process descriptions. The four primitive operations identified are FIND, ADD, DELETE, and MODIFY. The basic command format is: "OPERATOR (search criteria, ALL/UNIQUE, output, association, statistics)." These primitives are then used to build up three process categories:

1. read only -- FIND
2. modify -- FIND and MODIFY
3. size changing -- ADD or DELETE

For each process information requirements include precedence ordering, frequency of use, volume of data processed, and timing requirements in terms of both response time and when the data and process are available. According to the author, PSL/PSA provide a way of expressing and documenting the requirements from both perspectives. In terms of the actual design procedure for combining both perspectives, the author proposes for her doctoral dissertation to develop an initial design from the information perspective and modify it to meet the usage requirements.